A Review of Theoretical and Practical Challenges of Trusted Autonomy in Big Data

Hussein A. Abbass, George Leu, and Kathryn Merrick

Abstract

Despite the advances made in artificial intelligence, software agents, and robotics, there is little we see today that we can truly call a fully autonomous system. We conjecture that the main inhibitor for advancing autonomy is lack of trust. Trusted autonomy is the scientific and engineering field to establish the foundations and ground work for developing trusted autonomous systems (robotics and software agents) that can be used in our daily life, and can be integrated with humans seamlessly, naturally and efficiently.

In this paper, we review this literature to reveal opportunities for researchers and practitioners to work on topics that can create a leap forward in advancing the field of trusted autonomy. We focus the paper on the 'trust' component as the uniting technology between humans and machines. Our inquiry into this topic revolves around three sub-topics: (1) reviewing and positioning the trust modelling literature for the purpose of trusted autonomy; (2) reviewing a critical subset of sensor technologies that allow a machine to sense human states; and (3) distilling some critical questions for advancing the field of trusted autonomy. The inquiry is augmented with conceptual models that we propose along the way by recompiling and reshaping the literature into forms that enables trusted autonomous systems to become a reality. The paper offers a vision for a Trusted Cyborg Swarm, an extension of our previous Cognitive Cyber Symbiosis concept, whereby humans and machines meld together in a harmonious, seamless, and coordinated manner.

Index Terms

Big Data, Cognitive Cyber Symbiosis, Trusted Cyborg Swarm, Trust Bus, Trusted Autonomy, Trusted Autonomous Systems

I. Introduction

Trusted Autonomous Systems (TAS) are one of the grand challenges in this century. The proliferation of autonomous systems (AS) is a natural consequence of the opportunities they create for industry, health, education and the economy as a whole. These opportunities arrive with new technological and social challenges. TAS can be seen as a fusion of these challenges and acts as the umbrella that couples underneath it both objectives and challenges concurrently. Trusted Autonomy is the wider scientific endeavor to establish the science and engineering required to develop TAS.

For most of the literature, trust is seen as a human endeavor. It is seen to be owned by fields such as psychology [1], [2], social sciences [3], [2], [4], and organization sciences [5], [6], [7], [8]. Technologists and computational scientists have been modelling trust for a few decades [9], [10], [11], [12]. TAS brings this large spectrum of research together in the face of a new challenge: the creation of mutual trust between a human and a machine.

The above challenge carries forward classic research in human factors, automation, human computer interaction, and human robotic interaction to tackle the problem of how to get a human to trust a machine. However, in addition to this one-sided view of trust, the challenge is raising a more fundamental complex question on the other side of the equation: how can a machine trust a human?

This latter question is a socio-technological question on modelling complex systems, and forms the context of this paper. For a machine to trust a human, it needs to be able to 'proactively' sense the human, extract meaning and cues during sensing, manage identity to ensure it is dealing with the right human, estimate and manage human intent, and make an informed judgement on the level of trustworthiness of the human in a particular context in time and space.

Finding technological solutions to address this question is a complex endeavor that is compounded even more with the dynamic and possible chaotic nature of a trusting relationship. The slightest change in context may cause a dramatic change in trust estimation with a profound impact on trust dynamics. For example, a slight change in the security level of the communication channel between the human and the machine can cause a dramatic change in trust estimation.

To put it more succinctly, TAS require the machine to possess a high level of intelligence extending beyond the level of the task it is performing to include the level of the interaction it is involved with, the environment it is 'situated' and 'embodied' within, and to be aware of its own states, actions, motives, goals, and action production mechanisms. TAS require smart technologies for the machine to manage the interaction with humans; not just to convince the human that the machine is trustworthy, but to be 'self-aware' of trust to the extent of making its own decisions on the varying degree of trustworthiness of the human operators/users themselves during the interaction and based on context.

Designing the Artificial Intelligence (AI) models to manage the interaction between the human and the machine in the way explained above requires some fundamental building blocks. We structure this paper into three main sections.

Authors are with the School of Engineering & IT, University of New South Wales, Canberra-Australia, (e-mail: {h.abbass,g.leu,k.merrick}@adfa.edu.au).

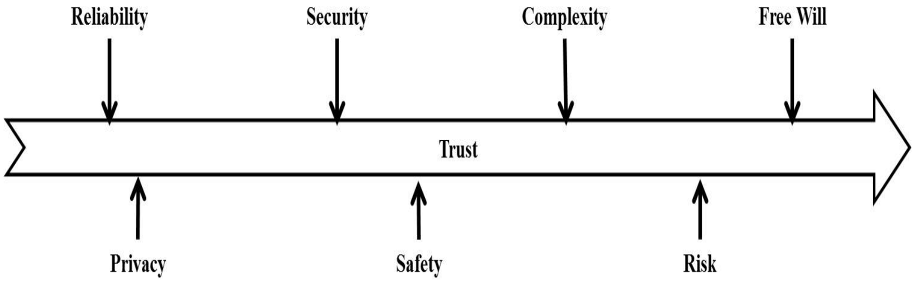

Fig. 1: Trust Timeline.

The first section (Section II) focuses on trust as the fundamental differentiator between classic AS research and TAS. We first start with the layers needed to define trust, then contrast how trust is modelled in the literature with these layers to offer two conceptual models: one is a computational structure model of trust and the other is a trust-based multi-agent architecture. We then define the 'trust aware' agent and present the concept of 'Trust Bus', which carries the information resultant from action production within the agent to the actuators. we adopt the term 'bus' in its computer architecture sense as in its classic use in computer architecture and service oriented architectures. The lens of the Trust Bus scopes the big data problem within this context to particular set of data and sensors. This scoping is essential as it guides the selection of sensors to be reviewed in the next section.

The second section (Section III) focuses on introducing big data to emphasize that the word 'big' is a relative concept defined in relation to the amount of processing available. We then review the sensor technologies available today that can facilitate the flow of information between humans and machines and augment the review with an overview of different processing concepts available to process these information. We conclude this section with a discussion on why the human-machine relationship is a big data problem.

In the third section (Section IV), we extend the discussion to the concept of autonomy and discuss the layers of autonomy. We then conclude this section by bringing together the first two sections and, first, offer the architecture required for an autonomous entity to exhibit both autonomy and trust, then second discuss the challenges associated with TAS from big data perspective.

II. Computations for and by Trust

A. The Road to Trust

Trust is a multi-dimensional concept. Research has established many facets and influential factors [12], [11] for trust dynamics [13]. We will discuss these facets and factors incrementally as each factor requires the list of factors listed before it to have a meaningful evolution of a trusting relationship.

The evolution model of trust presented in Figure 1 starts with the concept of reliability. A truster requires the trustee to be reliable; that is, the trustee needs to be consistent in its actions, values, and objectives during the lifetime of the relationship and stable in its performance. Fundamentally, a trusting relationship carries with it a level of risk; that is, the risk that the trustee will default. Reliability increases the confidence level associated with predictability. Therefore, it offers a better estimate of the risk a truster is exposed to during the trusting relationship.

Once reliability is established, trust improves through privacy. The order is a very important factor in this model. In the absence of any contextual information, a trustee that maintains privacy but not reliability is a high risk trustee. The reverse is not necessarily true as it depends on context. The truster requires assurance that private information in the trusting relationship are protected.

Privacy is a distinct concept from security. When a truster trusts a trustee, the trusting relationship involves transfer of certain data. For example, the trustee may need to know the identity of the truster and vice-versa. These data are communicated between parties for the purpose of fulfilling the trusting relationship. Any misuse of these data outside terms of the trusting contract is a breach of privacy.

Security, however, is more related to what is known as the CIA triad [14]: Confidentiality, Integrity and Availability. Confidentiality carries an element of privacy. A security breach where the database held by the trustee is broken into by an unauthorized third part can also be a breach of privacy. Maybe the third party only stole the trustee's data because it is not interested in the truster's data per se; thus, the truster data may not have been compromised. In this case, there is no privacy breach from the truster perspective but a clear security breach that may compromise the truster confidence in the trusting relationship.

Integrity breaches can be unintentional. A simple error in a piece of code can cause a deletion or modification of data and compromising the accuracy and validity of information. Availability is a function of the reliability of the IT service such that access is guaranteed whenever it is required and access control such that only authorized actors can access the data.

The above discussion can be seen from a reliability lens, where the concept of reliability discussed above is focused on reliability for delivering the service while the privacy and security elements relate to the reliability of the information assurance process.

The fourth factor is related to safety [15]. A truster needs to be safe in a trusting relationship. In a safety critical system, like air traffic management, the air traffic controller issues commands to pilots to ensure that the aircraft are separated (separation assurance). The pilot plays the truster since the pilot does not have access to all information related to the airspace and needs to trust the air traffic controller judgement. The air traffic controller is the trustee. The trustee is highly trained and reliable. The communication channel and data within this safety critical system are subject to strict information assurance procedures. But what will happen if the working environment for the air traffic controller is not safe? For example, there is a high possibility for a fire hazard, loss of power, etc. In this case, safety, or lack of in this example, can alter reliability of service and information assurance.

The four factors discussed so far are related to what we will call, technological imperfection. They are imperfect because we can't guarantee a 100% reliability, privacy, security and safety in any situation. Regardless of how much we will invest, the laws of dimensioning return would prevail.

The remaining three factors are related to what we will call 'cognitive imperfection' and start coming into play. The first factor is related to the complexity of the context for the truster and trustee. A truster would normally enter a trusting relationship to manage some level of complexity. Trust is a form of educated delegation. A truster would delegate to a trustee when the truster 'needs to'; that is, there is a benefit for the truster in trusting than self-performing the job. This delegation needs to reduce some form of complexity, be it, the technical complexity associated with performing the task, the time pressure on the truster to complete the task, or the increase in mental and cognitive complexity on the truster if the truster chooses to perform the task. As the level of complexity increases, the degree with which a truster trusts a trustee increases.

The second factor of cognitive imperfection is related to risk. As we mentioned before, a trusting decision involves a level of uncertainty associated with the possibility that the trustee defects [11], [13]. Risk within a trusting decision is both negative and positive. The expectations from the truster towards the good behavior of the trustee is a positive risk. It carries with it the negative risk resultant from the trustee defecting in the relationship. Taking high risk is an indication of high trust, while the reverse is not necessarily true.

The last factor in the evolutionary timeline of trust is 'free-will'; the fact that the truster is making the trusting decision not because of a necessity forced by an external actor who is attempting to exert its influence on the truster. With free-will arises full autonomy. An actor that is fully autonomous acts because it has the freedom to act. This freedom is not unbounded since social ties, social rules and norms, and interdependencies among actors in terms of resources and objectives limit the behavior of an actor. Free-will is the ability of the actor to make a decision within this bounded space autonomously and at its own discretion. This discussion begs the question of what is autonomy?

B. Computational Trust

In multi-agent systems, agents can cooperate to achieve goals that may not be attainable by individual agents. However, agents must manage the risk associated with interacting with others who may have different objectives, or who may fail to fulfil their commitments during a cooperative scenario. Models of trust offer a mechanism for reasoning about the reliability, honesty, veracity etc. of other agents, to provide a basis for determining whether cooperation is worthwhile. This section is a selective survey of computational models of trust that have been used in multi-agent systems, including the types of trust models, the agent-models in which they have been used, the type of cooperation achieved and the applications in which trusting agents have been used.

A number of existing surveys of computational trust have been made [16], [10], [17], [18]. Sabater and Sierra [16] classify trust and reputation models firstly as either cognitive or game theoretic. They also consider the information sources for trust, including direct experiences, indirect witness information and sociological information.

This section also considers these dimensions of trust models, but goes beyond this to survey the types of agent models in which trust models have been embedded, the applications for which they have been used and techniques for evaluating trust models. We also look beyond the scope of traditional agent-based systems literature to consider trust models from mathematical optimisation. The rest of this subsection is divided into two parts: trust modelling and computations by trust.

1) Trust Modelling: Computational models of trust are essentially mathematical functions that aim to represent certain aspects of philosophical or psychological theories of trust. The earliest computational models date back to the 1990s and are relatively simple statistical functions taking into account factors such as the situation of an agent, the utility of the situation and the importance of the situation. Later models have emerged in time with the development of more complex machine learning paradigms.

We first take the approach of Josang et al. [17], dividing the basic trust models into categories for statistical models, Bayesian analysis, discrete models, belief models, fuzzy models and flow models. However, we also extend this set of categories with a discussion of trust-regions from the literature of mathematical optimisation and computer networks.

- **Statistical Models:** An early approach to modelling trust was proposed by Marsh [19]. Marsh presents several definitions of trust that take into account different factors. First, he defines "the amount x trusts y" by $T_x(y)$. $T_x(y)$ has a value in the interval $[-1;1)$ (i.e., $-1 \leq T_x(y) < 1$). In his model, 0 means no trust, and -1 represents total distrust. He argues that the two are not the same since, the situation of 'no trust' occurs when the trustor has little or no information about the trustee, or is indifferent. In contrast, 'total distrust' is a proactive measure, requiring that the trustor reason about what it is doing. Marsh does not permit a value of 1 for trust, arguing that 'blind trust' implies that the trustor does not reason about the situation, and this violates philosophical definitions of trust. In a second definition, Marsh includes the notion of a situation α. A situation is a point or points in time relative to a specific agent. The importance of the situation to an agent x is written $I_x(\alpha)$. $I_x(\alpha)$ has a value in $(-1,1)$. The utility (cost or benefit) of a situation to an agent x is written $U_x(\alpha)$ also with a value in $(-1,1)$. The "situational trust" of agent x for agent y in situation α is then defined as:

$$T_x(y,\alpha) = T_x(y)U_x(\alpha)I_x(\alpha) \tag{1}$$

Estimates for each of these terms must then be obtained. Marsh assumes that values of $U_x(\alpha)$ and $I_x(\alpha)$ can be determined from the domain. Marsh suggests that the amount x trusts y at time t, might be determined as an average of situational trust values at previous times. Because agent memory is assumed to be bounded, one challenge is to decide which situations should be included in this average. Alternatives include all the situations the agent remembers or only situations that were similar to the current situation and so on. Different choices will clearly result in different trust values.

Other models have built on the work of Marsh, and maintain the idea of trust as a scalar value. Griffiths and Luck [20] define the trust in an agent y, to be a value from the interval between 0 and 1: $T_y \in [0,1]$. These numbers represent comparative values, and are not considered meaningful in themselves. Values approaching 0 are defined to represent complete distrust, and values approaching 1 represent complete or 'blind' trust. This is thus a simplified model, compared to that proposed by Marsh.

- **Bayesian Analysis:** More recent models take a Bayesian approach to modelling trust [21]. Rather than using 'situations' as the basis for calculation, binary valued interactions form the input for the trust calculation. An interaction $O_{x,y}$ takes a value of 1 if y fulfilled a contract with x, and 0 otherwise. 'Interactions' by this definition thus have a more constrained structure compared to Marsh's situations. The tendency of an agent y to fulfil or default on its obligations is governed by its behaviour, which is represented as a variable $B_{x,y} \in [0,1]$. $B_{x,y}$ specifies the intrinsic probability that x will fulfil its obligations during an interaction with y. That is, $B_{x,y} = Pr(O_{x,y} = 1)$. The level of trust $T_{x,y}$ is defined as the expected value of $B_{x,y}$ given a set of K outcomes $O_{x,y}^{1:K}$. That is:

$$T_x y = E(B_{x,y}|O_{x,y}^{1:K}) \tag{2}$$

In order to determine this expected value, a probability distribution defined by a probability density function (PDF) is used to model the relative probability that $B_{x,y}$ will have a certain value. An example is a beta PDF (see Equation 10 for example).
- **Discrete Models:** Other models stipulate a finite, discrete set of trust categories [22], [23]. For example, the model proposed by Abdul-Rahman and Hailes [22] stipulates four categories: very trustworthy, trustworthy, untrustworthy and very untrustworthy. An agent's trust degree is represented as a tuple, rather than a scalar value:

$$T = \{vt, t, u, vu\} \tag{3}$$

Trust is equal to the trust degree with the highest value. One agent's x belief in another agent y's trustworthiness in a given context α is represented as:

$$belief_x(y,\alpha,T) \tag{4}$$

An agent maintains this information for each other agent and context.
- **Belief Models:** A variation on discrete models is belief-based model such as those proposed by Josang [24]. In their model, trust is modelled as an opinion held by x about another agent y:

$$\omega_y^x = (b,d,u,a), \quad b,d,u,a \in [0,1] \tag{5}$$

where b represents belief, d disbelief and u uncertainty. a is the base-rate probability in the absence of evidence.
- **Fuzzy Models:** Fuzzy models provide another way to deal with the uncertainty associated with trust [25], [26]. A fuzzy set is a pair (X,m) where X is a set and $m : X \rightarrow [0,1]$ is a membership function that specifies the grade of membership of x in (X,m). For each $x \in X$, then x is not included in (X,m) if $m(x) = 0$. Conversely it is fully included in (X,m) if $m(x) = 1$. If we maintain our definition of x as an agent, then a fuzzy set (X,m) may be defined as very trusted agents, trusted agents, untrusted agents and so on.
- **Flow Models:** Josang et al. [17] define flow models as those that compute trust by transitive iteration through looped or arbitrarily long chains. The Advogato maximum flow trust metric [27], for example, computes trust of individuals in a group V relative to a 'seed' s of highly trusted group members. The input of the trust calculation are an integer number n, which represents the number of group members to trust, as well as the trust seed s. The output is a characteristic function that maps each member to a boolean value indicating trustworthiness:

$$T : 2^V \times \mathbb{N}_0^+ \rightarrow (V \rightarrow \{true, false\}) \tag{6}$$

The trust model underlying Advogato does not provide support for weighted trust relationships. Hence, trust edges extending from individual x to y express blind trust of x in y. Other examples of flow based metrics include the Appleseed trust metric [28].

A characteristic component of flow models is the seed (also called the trust root). The seeds of trust, are the positive assumptions about specific entities made by all entities in some community. The label authority is often applied to an entity that is the subject of a trust root. Models with formally modelled trust seeds may be termed centralized and those without may be termed distributed [29].

- **Optimisation Models:** Another approach to modelling trust has been used in mathematical optimisation [30]. This this field, a 'trust region' denotes the part of an objective function for which there is a good approximate model function. An example of a model function here is a quadratic surface. Algorithms that incorporate trust region methods are known as restricted step methods, and broadly take the following approach [31]: If a 'good' model of the objective function is found within the trust region, then the region is expanded. Conversely, if the model is poor, then the region is contracted. The quality of the model is evaluated by comparing the ratio of expected improvement from the model approximation with the actual improvement observed in the objective function. Thresholding of this ratio can be used as the criterion for expansion and contraction of the trust region. The model function is then 'trusted' only in the region where it provides a good approximation.

The models discussed above assume that trust is the same for all agents, essentially an objective model of trust. However, alternative models, such as that proposed by Marsh [19], discuss the role of the disposition of the trustor in computing trust. This interprets trust as a subjective concept. Marsh presents a spectrum of dispositions from optimists to pessimists. Optimists are more likely to trust, while pessimists are less likely to trust. In contrast to the generic, average-based definition of trust described above, he defines optimists to compute trust as in Equation 7, while pessimists compute trust as in Equation 8:

$$T_x(y) = max_{\alpha \in A}(T_x(y, \alpha)) \tag{7}$$

$$T_x(y) = min_{\alpha \in A}(T_x(y, \alpha)) \tag{8}$$

Again, the choice of which situations to include will influence the value of trust. Griffiths and Luck [20] use dispositions as a way of initialising trust values. In their model, trust values are initially inferred according to an agent's disposition: optimistic agents infer high values, while pessimists infer low values. Disposition also assumed to determine how trust is updated after interactions in their model.

An assumption of the models above is that trust values depend on factors such as the agent's disposition and its personal experiences with the trustee. When assessing trust in an individual with whom an agent has no personal experience, information must be obtained in another way. One method is to consult 'witnesses' to develop a collective opinion. This collective opinion is often referred to as 'reputation', and can be used as a factor in the assessment of trustworthiness. The following is a short review of computational models of trust that take this approach:

- **Statistical Models:** A simple model permits witnesses to report on a transaction by assigning one of three possible values 1 (positive), 0 neutral or -1 (negative). Reputation is computed simply as the sum or an average of all reported values over a certain time period [16]. More complex variants of this approach weight recent witness reports more highly than older reports, or stipulate different rates of reputation change depending on the current reputation of an agent [32].
- **Bayesian Analysis:** Still more complex variants take into account specific details of interactions between individuals to compute reputation. One such model [21] represents x's opinion of y's reputation as:

$$R^t_{x,y} = (m^t_{x,y}, n^t_{x,y}) \tag{9}$$

where $m^t_{x,y}$ is the number of successful interactions (interactions where $O_{x,y} = 1$) and $n^t_{x,y}$ is the number of unsuccessful interactions (interactions where $O_{x,y} = 0$) experienced by x. x may communicate this opinion with a third agent z. The communicated opinion $\hat{R}^t_{x,y}$ may be distinguished from the true opinion $R^t_{x,y}$ as is possible that x may not report accurately for a range of reasons. z can develop a collective opinion of y by summing values of $\hat{m}^t_{x,y}$ and $\hat{n}^t_{x,y}$ collected from multiple other agents x and using these values to calculate the shape parameters for a beta distribution, as in Equation 10.

$$\begin{aligned} & N_{z,y} = \sum_x \hat{n}_{x,y} \quad and \quad M_{z,y} = \sum_x \hat{m}_{x,y} \\ & \alpha = M_{z,y} + 1 \quad and \quad \beta = N_{z,y} + 1 \\ & f(B_{z,y}|\alpha, \beta) = \frac{(B_{z,y})^{\alpha-1}(1 - B_{z,y})^{\beta-1}}{\int\limits_0^1 u^{\alpha-1}(1-u)^{\beta-1}\mathrm{d}u} \end{aligned} \tag{10}$$

The final value for trust is calculated by applying the standard equation for the expected value of a beta distribution to these parameter settings: $T^t_{z,y} = E(B_{z,y}|\alpha, \beta) = \frac{\alpha}{\alpha+\beta}$. Then, z may also include their own opinion in the trust calculation if such data exists. If all agents have complete information about the true opinions of all other agents, then it is said that a global or objective measure of trust results. Otherwise, trust remains local or subjective [28].

- **Network Models:** Another set of trust models based on witness data have been proposed for use in computer networks, and peer-to-peer computing, particularly in applications where multi-agent systems are responsible for transporting valuable data over a network. Examples of trust models for computer networks include PeerTrust [33], FCTrust [34] and SFTrust [35]. Other models build on this work [33]. They use parameters such as the feedback a peer obtains from other peers (also called satisfaction); feedback scope (such as the total number of transactions a peer has with other peers); the credibility factor for the feedback source; the transaction context factor for discriminating critical from less critical transactions; and the community context factor for addressing community-related characteristics and vulnerabilities. An example of a network trust metric based on these parameters is:

$$T(x) = \alpha \sum_{i=1}^{I(x)} S(x,i)Cr[p(x,i)]TF(x,i) + \beta CF(x) \tag{11}$$

  $I(x)$ denotes the total number of transactions performed by peer x with all other peers. $p(x,i)$ denotes the other participating peer in peer x's i^{th} transaction. $S(x,i)$ denotes the normalised satisfaction peer x receives from peer $p(x,i)$ in its i^{th} transaction. $Cr(.)$ denotes the credibility of feedback, $TF(x,i)$ denotes the adaptive transaction context factor for peer x's i^{th} transaction. $CF(x)$ denotes the community context factor. α and β denote the normalise weight factors for the collective evaluation and the community context factors.

The models we discussed thus far are static, that is, they do not model the change in trust over time. Marsh [19] also presented a notation for change in trust as an update function for Equation 1:

$$T_x(y,\alpha)^{t+1} = T_x(y,\alpha)^t + \Delta \tag{12}$$

Marsh [19] considered change in trust over time as a result of different dispositions. For optimists, Δ is a large positive number in response to situations with positive utility, but a small negative number in response to situations with negative utility. In contrast, for pessimists Δ is a small positive number in response to situations with a positive utility, but a large negative number in response to situations with negative utility. Different types of optimists and pessimist can be created by varying the choices of Δ. Jonker and Treur [36] expand on the two trust dispositions by distinguishing six types of trust dynamics: blindly positive; blindly negative; slow positive + fast negative; balanced slow; balanced fast; and slow negative + fast positive. Other models also assume that trust may change over time due to progressive accumulation of experiences. In this case, Equation 2 becomes:

$$T_x(x,y)^t = E(B_{x,y}|O_{x,y}^{1:t}) \tag{13}$$

Change in trust over time, is generally assumed to be a result of information gathering 'experiences'. A parallel concept that is often discussed as a result of such experiences is confidence [20], [21], [25]. The intuition is that the more evidence used to compute trust should result in higher confidence in computed trust values. Confidence is thus not a direct measure of trust, but an associated concept that tells an agent something about its trust calculation.

An example of a confidence metric that can be derived from a Bayesian definition of trust defines confidence $\gamma_{x,y}$ as the posterior probability that the actual value of $B_{x,y}$ lies within an acceptable margin of error ϵ about $T_{x,y}$. That is:

$$\Gamma_{x,y} = \frac{\int_{T_{x,y}-\epsilon}^{T_{x,y}+\epsilon} w^{\alpha-1}(1-w)^{\beta-1}\mathrm{d}w}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\mathrm{d}u} \tag{14}$$

An alternative, simpler definition of confidence used in a fuzzy setting is a simple sum of the number of experiences (positive and negative) encountered by the agent [25].

Reliability is an associated concept with trust and confidence. Reliability theory is the study of the performance of a system of failure-prone elements [29]. Reliability is studied in many contexts, including software reliability, network reliability, hardware reliability and so on. Reliability is concerned with the ability of computer systems to carry out a desired operation. Reliability has been understood in different ways by different researchers with respect to trust. Some researchers have equated reliability with trust [29], while others make the weaker assumption that reliability influences trust [37].

In summary, different trust models have been proposed over the last two decades, with new models emerging in line with new developments in machine learning theory. The next section examines the agent architectures in which these models have been used.

2) Computations by Trust: Agents are entities that sense their environments using sensors, reason about data so obtained using one or more reasoning processes, then act to change their environment using effectors. While some theoretical work on trust models trust in the absence of a specific agent architecture, there are some common agent frameworks that have been used to examine the models. Much of the literature on trusted agents makes the assumptions that agents are part of an open multi-agent system (MAS), in which individual agents are self-interested. That is, each agent has a goal or goals, but agents in the MAS do not necessarily share a common goal. This is considered a key reason for the necessity of trust. Aside from the assumptions of goals and self-interest, trust models have been used in a range of agent architectures, including belief-desire-intention (BDI) agents, reinforcement learning agents and motivated agents. These architectures, and the way in which they incorporate aspects of trust, are reviewed as follows:

- **Belief-Desire-Intention Agents:** The BDI model [38] enables the definition of intelligent agents that have:
  - Beliefs about the world,
  - Desires that they would like to achieve,
  - Intentions, subsets of desires to which they have made some commitment,
  - Plans that, if successful, will achieve intentions.

  The BDI architectures is one of the longest-standing models of intelligent agency used in MASs. It has thus been a popular target for the exploration of trust models [39], [40], [41]. One example is the ability-belief-commitment-desire (ABCD) model of trust [39], which integrates with the logic of rational agents (LORA) BDI agent theory. Koster et al. [41] provide a methodology for incorporating multiple different trust models BDI agents.
- **Motivated Agents:** Griffiths and Luck [20] extend a BDI framework with motivations. In addition to the traditional components of the BDI model however, Griffiths and Luck [20] argue that motivation is an extra component required to achieve true autonomy in such agents. Motivations are high-level desires that characterise an agent; they guide behaviour and, at a fundamental level, control reasoning. They argue that short-term teams are not appropriate since motivations are too dynamic when considered over a short period of time, i.e. over a small number of tasks; in the short-term agents' motivations may be out of step resulting in agents' failure to share common goals at the same time. Similarly, long-term coalitions are unsuitable since although a given agent has a fixed set of motivations, the general trend of which motivations are active may change.
- **Learning Agents:** Learning agents can modify their internal structures in such a way to improve their performance with respect to some task. Different approaches to learning existing, including learning from examples (supervised learning) and learning from reward and punishment in response to interactions with an environment (reinforcement learning). Many agents that incorporate models of trust are learning agents, because they accumulate experiences (examples) and change their interaction partners over time to improve their performance at some task. However, some approaches also use secondary learning algorithms to augment the trust model. Aref and Tran [26], for example, describe a fuzzy trust model in a Q-learning reinforcement learning framework to help trust evaluating agents select beneficial trustees for interaction in uncertain, open, dynamic, and untrusted MASs. Experimental results indicate that the proper augmentation of a fuzzy trust subsystem to Q-learning can be useful for trust evaluating agents, and the resulting model can respond to dynamic changes in the environment.
- **Multi-Agent Optimisation:** Trust has also been studied in optimisation settings such as particle swarm optimisation (PSO). In PSO, agents (particles) represent solutions to a problem. Agents move around in a search space attempting to discover and converge on optimal solutions to the problem in question. Xu et al. [42] proposed a variation of the PSO algorithm for trust path selection in networks. They do not define a model of trust, but rather a novel PSO algorithm that can embed a given model of trust. Huang et al. [43] propose a different variation of PSO for incorporating trust to solve grid task scheduling. PSO algorithms are also particularly compatible with the trust-region formalism, and examples of algorithms that incorporate trust-regions with PSO have been proposed [44].

The examples above demonstrate that trust models have been developed across a number of different agent frameworks. These include learning, optimisation and planning architectures. Trust models have been particularly well studied in BDI agents. Currently we see that the trend is to develop different kinds of trust models for use in different settings. That is, there is currently no universally accepted trust model appropriate for a range of different types of agent frameworks. However, there are some trust models that have been used and adapted in several different settings. Marsh's early work [19] is one such example. These models are general enough for adaptation to specific agent frameworks or application domains.

Applications of trust models include a range of abstract scenarios modelled in simulation, as well as a number of well known, real-world applications. Abstract applications include delegation, cooperation and detection of deceitful agents. Real world applications include a range of internet mediated service providers. A number of applications are discussed below

- **Delegation:** Trust models can be used to determine whether a task should be delegated to another agent. That is, trust is used by agent x to decide if it should allocate a certain task to agent y [45].
- **Cooperation in Multi-Agent Systems:** Trust can also be used to determine whether multiple agents should cooperate. Griffiths and Luck [20] describe different types of cooperative groups that can occur between agents:
  - Teams: short-term cooperative groups formed to achieve a specific goal. Agents hold a common goal at the time of

forming the team, and expect immediate benefit from achieving the goal.

  - Clans: medium-term cooperative groups that enhance individual goal achievement, with increased group performance a useful side effect.
  - Coalitions: long-term cooperative groups formed to achieve a specific goal, which can be broken into sub-goals. Agents may have different preferences for these sub-goals, but are willing to forgo immediate individual benefit in the interests of long-term benefit for the coalition.

  Marsh [9] outlines different ways in which trust plays a part in the formation of cooperative groups:

  - Trust may influence whether an individual desires to join a group.
  - Trust may influence whether a group desires an individual to join them.
  - Trust may influence how members of different groups interact with members of other groups.

- **Detecting Deceitful Agents:** Deceit is considered commonplace in certain industries such as trade [46], [47], [48], [49], computer networks [50] and online communities [33]. Understanding deceit is vital for detecting it, and for the design of governance mechanisms to discourage deceit. Observed regularities related to trust can be used to form the basis of models for deceit. This can assist with the detection of deceit in trade scenarios [46], [47], [48], [49], or malicious behaviour on computer networks [33], [50].
- **Reputation:** Reputation models are increasingly ubiquitous among internet mediated service providers [17]. For example, eBay (www.eBay.com) uses a reputation model to rate the trustworthiness of sellers. AirBnB (www.airbnb.com.au), use reputation models to rate the trustworthiness of hosts.

An appropriate use of trust models necessitates mechanisms for evaluating the performance of these models. Yu et al. [18] include in their survey of trust models a survey of performance assessment methods for such models. They note two major approaches used in the literature:

1) Simulation-based evaluation; and
2) Evaluation through real world datasets.

The approaches have been applied either individually or in combinations by researchers. They found that the most widely used evaluation method is via simulations. An example of a simulation testbed is the agent reputation and trust (ART) testbed [51], which simulates painting appraisers with varying levels of expertise in different artistic eras. Clients request appraisals for paintings from different eras. Appraising agents may also request opinions from other appraising agents. Appraisers receive more clients, and thus higher profits, for producing more accurate appraisals. The testbed assesses the quality of appraisers' trust-based decisions objectively by the profits they achieve, as well as the accuracy and consistency of their appraisals.

Simulations permit researchers to vary experimental conditions. For example, they may simulate different ways in which a trust model may be exploited. In contrast, real world data may enable researchers to have a better idea of how their models would work under realistic environment conditions. However, because many datasets are not specifically collected for the purpose of evaluating trust models, they may lack the ground truth about the user behaviour to facilitate more in-depth analysis of the performance of proposed trust models [18].

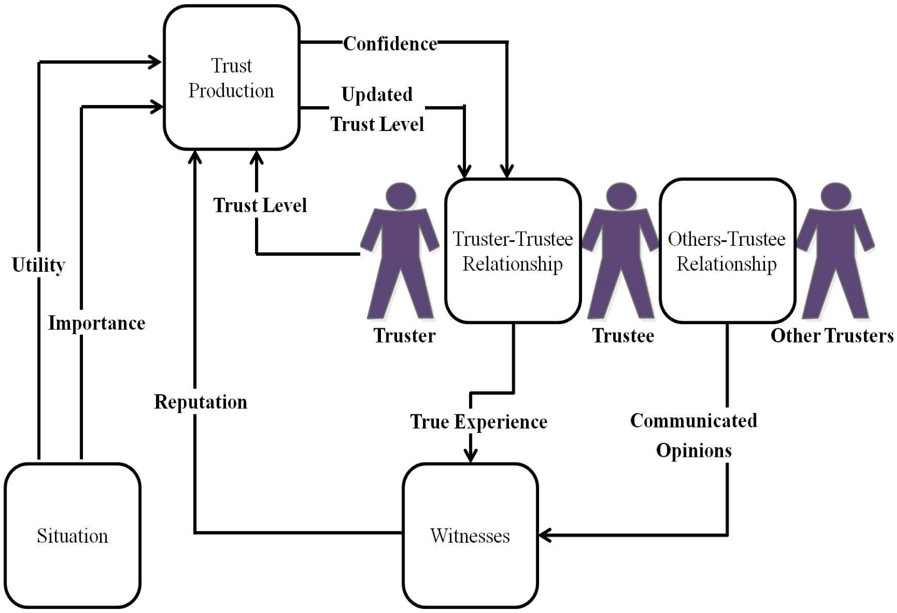

Fig. 2: A schematic diagram of different components of a trust model based on the literature reviewed above.

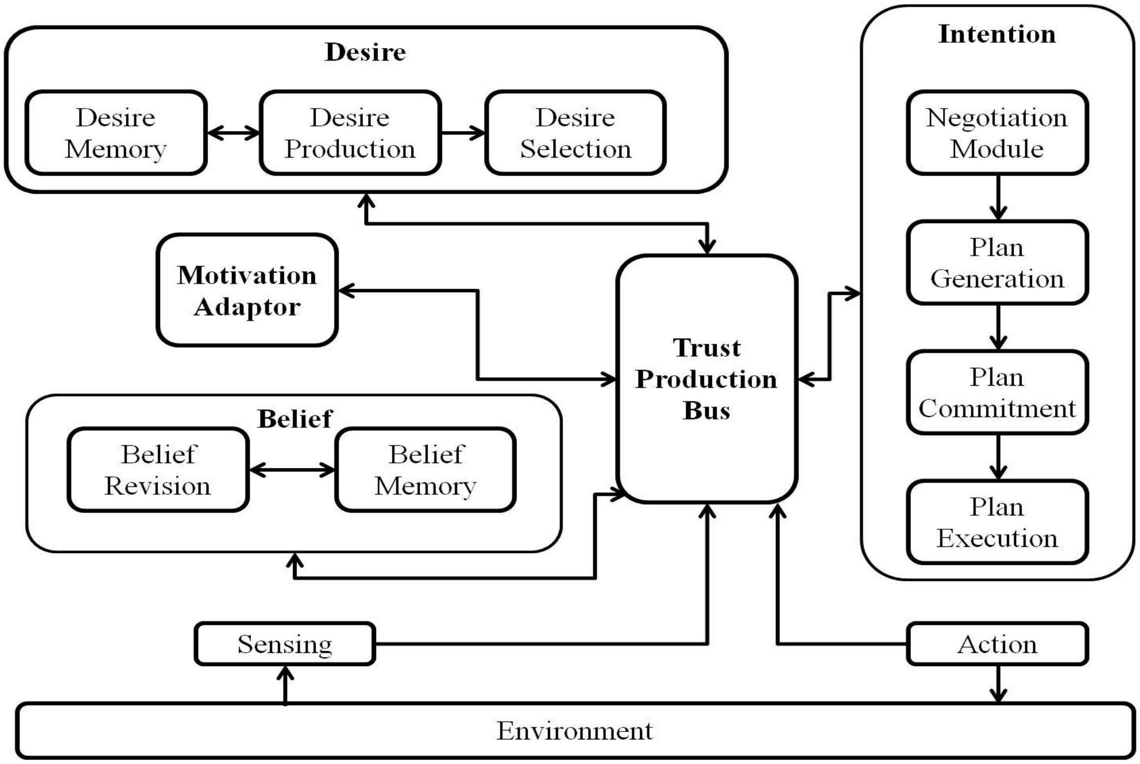

Fig. 3: A schematic diagram of a modified BDI agent.

C. *Synthesis of Computational Trust*

This section has considered four aspects of trust models:

- The inputs, processing types and outputs of trust models,
- The agent architectures in which trust models have been used,
- Applications of computational trust,
- Techniques for the evaluation of trust models.

Regarding the inputs, processing types and outputs of trust models, a key division is the direct and indirect input types. Indirect input is frequently indicative of a reputation based model of trust. Trust has been incorporated extensively in BDI agent architectures, but has also been examined in motivated agents and learning agents. Abstract applications include delegation, cooperation and detecting deceit in MASs. Concrete applications occur particularly in internet mediated service industries, which tend to make use of reputation models and computer networks for security purposes.

Figure 2 presents a compilation of the literature on trust modelling that was reviewed in the previous section.

This diagram clearly identifies the four main inputs to a trust model. One input depends on the truster's initial belief about the level of trust it holds towards the trustee. One input depends on direct and indirect experience encounters with the trustee. Two inputs depend on context and represent the importance and value of the context.

The truster needs to obtain two indicators from the trust model: the updated level of trust and the model confidence in this update. The truster may choose to ignore an estimate with low confidence.

This model is intuitive, simple and basically captures the basic building blocks from across the wide literature on modelling trust. Variations can exist by expanding some of the elements. For example, how to calculate the importance of a situation or what personal attributes of the truster and trustee we can use for trust update. Another example is related to the risk appetite and attitude of the truster, which may influence the truster's trust-update functions. All these are possible, with their utility dependant on data availability and the cost-benefit trade-off that needs to be taken into account when deciding on how complex a trust-model should be.

Trust production is the set of mechanisms needed to make a trusting decision. Figure 2 conceptually depicted one input-and-output view of a trust production system. It is equally important to know where trust sits within the wider agent architecture. We will start with the BDI architecture since we reviewed it above, but we will then move high up in the abstraction hierarchy to offer system-level architectures that integrate trust production more profoundly than the simple equation model used in almost all of the literature we reviewed on trust.

The amalgamation of trust production with BDI necessitates a new perspective on BDI that explicitly acknowledges the complexity of trust production instead of merely reducing it to the equation-based approach. We compile the above discussions

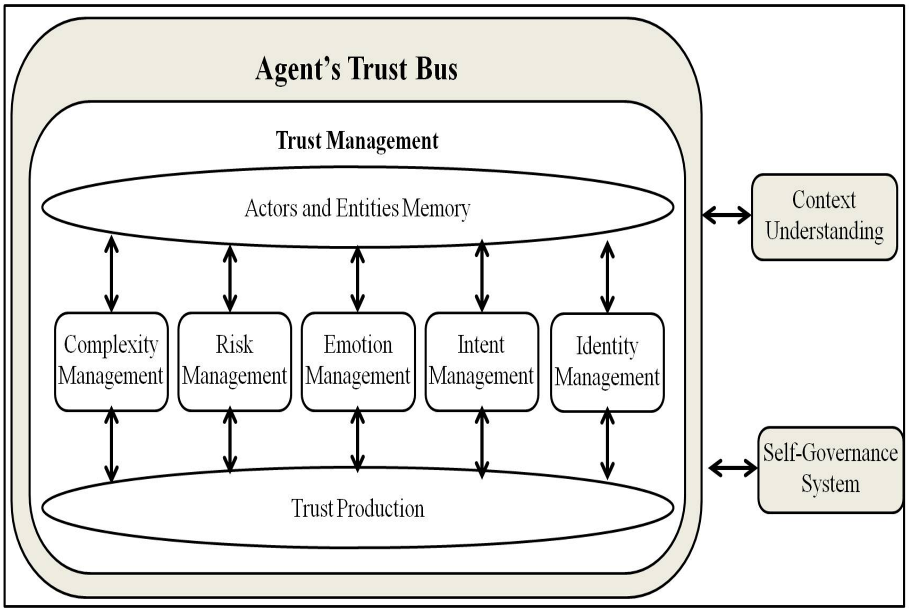

Fig. 4: TrustBus.

into the Belief-Desire-Intention-Trust-Motivation (BDI-TM) architecture as shown in Figure 3.

This architecture acknowledges two additional components to BDI; these are: Trust Bus and motivation adaptor. The Trust Bus interacts with the belief component to estimate trustworthiness of trustees but also produces information which are essential for the belief update mechanism: the primary learning mechanism in a BDI architecture. The Trust Bus acts as the message passing interface that learns – through automatic knowledge acquisition methods [52] – from and modifies massages as they get transferred from one system module to another.

The second component is related to motivation, which acts as an adaptor to desires. We prefer in this architecture to make motivation as a stand-alone component, simply because it is an adaptor to the overall desire production box, not just the desires themselves and can influence, and be influenced with, trust production as well. Beliefs memory and trust need to get fused with motives before motives adapts desires.

The previous discussion begs question on the constituents of the Trust Bus, and whether this bus is a simple message-passing interface or does it do more than that?

Figure 4 expands the Trust Bus into its constituents: Actors and Entities Memory, Trust Production, Identity Management, Intent Management, Emotion Management, Complexity Management and Risk Management. Each of these components deserves a brief overview below.

The literature on trust implicitly emphasized the need to be aware of the actors in the environment (at least the trustee) and other entities. The Actors and Entities memory maintains the information stored on these actors and entities for retrieval when needed. The identity management module is responsible for identification of actors and continuous authentication of their identities while interacting with the system. This is a critical component underpinning the mechanisms needed to manage reputation, reliability of information, privacy and security.

The intent management module acts as a smart engine to objectively estimates actors intents and use these estimates to offer informed advice to the trust production module on the reliability and value of these intents. It also offers what-if services to the trust production module to explore the space of possible implications of trust.

The three remaining modules represent the cornerstones for trust in psychology and social science research and were elicited in our previous work [11], [13], [12]. These are emotion, risk, and complexity. Emotion management models and learns affective states of other actors in the environment. An advanced version of this module can manage the emotion states expressed by the AI when needed. The risk management module maintains a risk registry, analyzes uncertainties and offer what-if services to the trust production module. The complexity management module estimates the cognitive and task complexities in the context of other agents and the truster. The literature on trust shows that complexity is a critical driver for trusting decisions. The truster can get exposed to a level of complexity forcing unwise trusting decisions if this complexity is taken for grant. The complexity management module analyses the complexity surrounding the agent in a given situation, including the complexity surrounding other actors in the environment and their abilities to manage it.

The Trust Bus is used to scope the discussion on the nature and types of sensor technologies required by the Trust Bus to estimate human and group states. The Big Data nature of this problem warrants a special treatment and raises both theoretical and practical challenges. This topic will be the context of the next section.

III. Big Data at the Human-Machine Interface

The machine needs to be able to sense humans, their behavioral attributes, cognitive states, actions and decisions, and even the rational and explanations they make about their own judgements. This sensing is key for the machine to acquire the necessary information it needs to make a judgement about a human. It is very similar to human-to-human interaction, where humans actively and subconsciously sense each others facial expressions, body language, voice tone, and even skin temperature during a hand shake or a hug. With today's technologies, such as wearable devices, the machine can access even more information about the human's psychophysiological states that the humans themselves may not be conscious of. Today, Google applications can map out humans' daily routines, food and cloth preferences based on their shopping style, estimating probability of heart attacks based on their life style, social networks based on people in their social circuit, similar locations at night, similar routes, and more.

In addition to sensing the human, the machine needs to sense the environment. A change in human body temperature can simply be attributed to changes in the air conditioning setting. A change in a human voice signal can be attributed to the temperature in the environment, interferences between something in the environment and the microphone, or simply a result of a long speech or fatigue. Without sensing the environment, the machine is unable to attribute changes in the signal that it is sensing to different possible human states.

Sensing is a pre-requisite for any modelling that a machine needs to estimate a human's trust level. This section covers this topic through a Big Data lens. Before we regress further, we start the discussion with how big is a big data problem.

A. *How Big is Big*

How big is big? This is usually the question that gets asked by those who are skeptical about the word 'big' in big data. However, the word 'big' is not merely about volume. It reflects the size of different characteristics that together form the basic features that one can use to characterize a problem as a big data problem.

The common characteristics for big data are summarized in different words that start with the letter 'V'. This generated an evolution of these characteristics starting with the 3Vs model representing volume, velocity, and variety, then the 5Vs with the addition of veracity and value, and finally the 6Vs with the addition of variability [53], [54], [55].

Each of the Vs represent one characteristics that by itself, the problem may get called a big data problem, but the challenge gets more complex when multiple Vs coexist together. Naturally the first V represents the volume or size of the data. A big data problem may have a significantly large number of features or simply terabytes or petabytes files.

The second V stands for velocity, or the speed with which the data change. Such changes may cause movements of the decision boundaries for a decision tree, alterations of the degree of nonlinearity in a regression model, or variation in the prototypes and numbers of clusters in a clustering problem.

Variety, or heterogeneity, refers to the distributed nature of the data. A customer may have different forms of records across different databases. Some of the records may take the form of buying patterns, while others may take the form of voice signatures to authenticate the customer during online banking, or unstructured data collected during an interview with the customer.

Veracity is related to the amount of noise that may exist in the data and the level of trustworthiness of the source. Veracity can further be divided into accidental veracity and intentional veracity. The former represents the classic noise in a communication channel associated with natural and/or physical causes. The latter represents intentional noise caused by a malicious act. These two factors play an important role in the validity and authenticity of any conclusion drawn from a dataset.

The second last V is the value characteristic; capturing the value for money and worthiness of the decision or conclusion drawn from the data. This is a cost-benefit-analysis question. Low cost data can generate high benefits, while high cost data may be too costly for the benefits they generate.

Last, but not least, variability represents the dynamic nature of most IT infrastructure. Data exchange protocols change. The architecture of a database changes. The data format received from third parties change. These changes are necessary as we optimize the databases, customize the data differently for different customers, and so forth. Nevertheless, these changes impose major challenges in big data. The more variability that exists in the problem at hand, the harder the big data problem is.

Autonomous systems are at the heart of the Big Data problem. The problem of autonomy is that it can generate unpredictability in the environment. As a software robot moves from one computer to another, protocols may change, data format may change, data worthiness may change, source reliability may change, and so forth. An autonomous entity has limited processing power relative to the data it encounters during its operations. Autonomy is a challenge for big data, and equally though, big data is a challenge in autonomy.

Adding trust to the equation creates the biggest challenge overall. Trust glues autonomous systems. This glue is not a one-off glue. Trust is dynamic and requires dynamic strategies to manage it in an autonomous system.

B. *Sensors: What can they offer the human machine problem*

Sensing is what allows a system to interact with its environment in a flexible and non-deterministic way, as apposed to performing preprogrammed actions based on a set of scheduled functions and/or rule sets [56]. In other words, sensing is a fundamental mechanism and condition for a system to be able to operate autonomously in order to fulfil its purpose. Recent years have marked unprecedented advances in sensor technology, making available miniaturised, highly portable, accurate and reliable sensors, with high throughput and low cost [57]. This allowed the resultant sensor-based autonomous systems to embed more sensors, which in turn allowed them to operate in more complex and more dynamic environments. On the one hand, one can consider that the better the sensing is, the more data the system gathers from the environment, and further, the more informed and accurate the resultant decision is. However, on the other hand, as the amount of data generated by sensors grows, so do (1) the amount of processing the system needs to perform in order to generate a useful decision, and (2) the difficulty of taking the decision given the increased amount of choice. Rich data coming from numerous and high quality sensors could be simultaneously an enabling and a disabling factor for the operation of autonomous systems, with respect to the processing power available to the system.

We can give the example of an autonomous unmanned vehicle, which must deal with data coming from the activity of human agents served by the vehicle (e.g. passengers of an autonomous car, or passengers of an aircraft), and also from the surrounding technological agents (internal interaction with own mechanical sub-systems or external interaction with other vehicles participant in traffic). Another example can be an autonomous system for patient diagnosis and monitoring in a hospital. This needs real-time and historical data from the respective patient and historical data from numerous other patients in order to ensure statistical validity of the decision. Also, it needs data from the various machines and equipment operating in the hospital.

Regardless of the source and type of activity, modern sensors operate in a digital manner, and generate data through sampling of their sensing. While sensing and processing of sensory information are taken for grant for biological agents (e.g. a human observer/decision-maker), it is not the case with artificial autonomous systems, which operate through digital computation and perceive everything as data. Thus, it becomes essential to have a clear view of the existing sensor technologies especially with a focus on the amount of data they generate as part of their normal operation. At the same time, it is essential to have a clear view on the existing processing/computing technologies and architectures in order to understand the potential limitations the autonomous systems can experience when they deal with the big data coming from their sensory-subsystem.

1) Sensing human activity: Human activity sensors have undergone unprecedented development in recent years. Sensors and sensing systems that were once reserved for clinical or research applications are now easily accessible to the general public; are manufactured and commercialised on a large scale; and are being used across multiple domains in versatile and low cost applications. Patel et al. [58] identified the key facilitators that enabled this explosion of achievements in sensing human activity. The authors discuss the sensing systems in terms of three essential building blocks: (1) the sensors, i.e. sensing and data collection system (2) the communication system needed to transfer data to the processing centre, and (3) the data analysis techniques that extract the relevant information from the raw data. A refinement of the three component view presented in [58] was later discussed by Mukhopadhyay [59], who considers eight components where sensing research should concentrate for improving the overall performance of human activity wearable sensing systems: (1) types of sensors to be used, (2) type of communication protocols to be employed, (3) type of activities to be monitored, (4) techniques for extracting important features from activities, (5) design and development of small, light-weight, powerful and low-cost smart sensor nodes, (6) harvesting of energy for normal operation and communication, (7) ability to be used with the present day mobile devices, (8) potential to be reconfigured for new applications/users.

Both studies, [58], [59], note how each of these components have been on a path of miniaturisation, where the advances in micro/nano-electronics and related technologies allowed the resultant systems to become smaller and more powerful, and in the same time more energy efficient. These technological advances further allowed the emergence of fully wearable sensing systems that provide versatility and flexibility in both design and operation; in addition, they allow long-term non-intrusive monitoring and data collection. In [59] the author states that this surge of wearable sensing will continue to expand towards more and more domains, and cites industry reports concluding that the penetration of research advances towards the consumer market will lead to an estimated market growth to a value of over $20 billion by 2017. Wearable sensing systems have been extensively used in domains such as sports and recreation [60], human-computer interaction, work-space and ergonomics, games and entertainment, and last but not least, education.

Numerous reviews discussed sensors and sensing systems in all these domains from the points of view of their component blocks and subsequent technology, with a focus on the aspects that allowed miniaturisation and wearability. Extensive and very recent reviews can be found in [58], [61], [62], [63], [64], [59] and [65]. However, the resultant amount of data was not discussed so far in the literature. In the following, we describe the most common types of sensing systems used in monitoring human activity [59], and we focus our discussion on the sampling rate and the subsequent data resulting from their operation.

- **Body temperature:** Body temperature is one of the most common physiological parameters used for monitoring human activity, usually measured by wearable sensors attached to the skin. The body temperature in different parts of the body, and variations of it can give an indication of symptoms of medical stress that signal the existence, or the possibility of acquiring various health conditions. Body temperature monitoring can be used either for determining the respective physiological/medical conditions or for classifying and storing for further use historical physiological activity data, or even for harvesting energy from body heat [59]. Traditional measurement of body temperature was in the form of one-off measurements, for clinical purposes. However, continuous monitoring is nowadays common, especially as part of wearable activity monitoring devices and systems. In Table V all devices listed as incorporating temperature sensors, refer to skin temperature, and operate through contact to skin tissues typically in the wrist area, where wristband or watch type activity monitors are attached. Contact technologies such as thermocouples, resistive temperature detection, and organic diodes have been proposed over time [66]. Recently, contactless temperature sensors based on infra-red waves have been developed for accurate and hygienic monitoring of temperature in various parts of the body, such as FT90 contactless clinical thermometer from Beurer [67].

- **Heart rate:** Heart rate (HR) is also one of the most common physiological parameters monitored using wearable sensors for various purposes. Heart rate is a precisely regulated variable that is critical for diagnosing various medical conditions and disease of human, however its monitoring is also essential in sports and entertainment. A variety of sensor modalities are available for heart rate measurement and monitoring, such as sound based [68], [69], optics (photoplethysmography) based [70], [71] or pressure based [72]; also monitoring by visual cues such as the face colour [59] have been also reported in the literature. Historically, the early wearable HR monitoring devices were pressure-based and used chest-straps to attach to upper body, e.g. Garmin HRM-Tri and HRM-Swim [73] dedicated to triathlon and swimming athletes, respectively. Recently, the due to the high demand on the wearable consumer market, technology migrated towards optic-based devices, which are currently implemented on all wristband, smartwatch, pendant or garment activity sensors. In Table V all activity sensing devices that include HR monitoring are LED-enabled optic-based sensors. An in-depth discussion on the differences between the two technologies, and the trends in wearable HR devices is provided in [74].
- **Electroencephalography:** Electroencephalography (EEG) is an electrophysiological sensing method that captures brain activity in the form of brainwaves coming from the aggregation of the electrical neural impulses; thus, EEG is based on electromagnetic wave sensing. Typical EEG sensing systems consist of several electrodes that measure in different regions of the brain the voltage fluctuations resulting from ionic current within the neurons. The sensing can be invasive, where the electrodes are implanted under the scalp, or non-invasive, where the electrodes are placed on the scalp either individually or as part of customised caps [75] in which the electrodes are precisely positioned. However, with the recent advances in the transducing elements of EEG devices, intrusive methods are now less and less used, while wearable caps and the very recent dry sensing headsets are becoming more common. EEG sensing was traditionally used in clinical applications, such as the traditional diagnose method for epilepsy, but also in supporting the diagnosis, monitoring, and treatment of other brain conditions, such as sleep disorders, coma, encephalopathy, brain death, etc. Research applications are especially in the fields of cognitive science and engineering [76], and human/brain-computer interaction [77], where the high temporal resolution of EEG sensing brings valuable information about real-time brain activity, capturing changes in cognitive load and states. High temporal resolution is the main advantage of EEG over a number of other brain imaging and brain activity sensing methods, which include magnetic resonance imaging (MRI) and functional MRI, single-photon and positron emission tomography, magnetoencephalography (MEG), nuclear magnetic resonance spectroscopy, electrocorticography, near-infrared spectroscopy, and event-related optical signaling. Classic clinical EEG sensing systems operate at sampling rates between 100 and 256 Hz [78], [79], [80], while some of the latest EEG systems are capable of recording at sampling rates as high as 20 kHz [81], [82], [83], that is, a sub-millisecond temporal resolution. However, while this temporal resolution is the main advantage of EEG sensing, it is also one of the challenges for using it in real-time applications, given that EEG systems can record in parallel on as many as 256 channels. In Table I we summarise the main EEG equipment available to date, and list for each of them the essential sensing data-generator features.
- **Eye movement sensing:** Sensor systems that capture eye movement are used in clinical contexts primarily in ophthalmological diagnosis and in research contexts for eye tracking applications mainly related to human factor, cognitive science and engineering, and human-computer and brain-computer applications. The common eye movement sensing systems are either based on electric sensing or optical sensing.

  The most common electric sensing method is Electrooculography, which measures the corneo-retinal standing potential existing between the front and the back of the human eye. To measure eye movement, pairs of electrodes are placed at the opposite sides of the eyes (i.e. either above and below the eye or to the left and right of the eye). If the eye moves from centre position toward one of the two electrodes, this electrode "sees" the positive side of the retina and the opposite electrode "sees" the negative side. The resulting signal is called electrooculogram, in which the difference in potential occurring between the electrodes is recorded as a measure of the eye's position. EOG captures eye movement only, while more complex electric sensing methods, such as the Electroretinogram, can also capture and monitor eye response to individual visual stimuli.

  The optical eye movement sensing typically consists of eye-tracking systems based on specialised video cameras that capture retinal reflection and refraction as result of eye's exposure to visual stimuli.
- **Electrocardiography:** Heart activity can be also measured and monitored using sensing systems. The typical sensing method for heart activity monitoring is electrocardiography (ECG), which uses electrical sensing for detecting the electrical changes that arise from the heart muscle depolarising during each heartbeat. These changes are captured on subject's skin using electrodes placed on relevant locations on the body. Typical ECG systems, used in clinical contexts, consist of ten electrodes placed on patient's limbs and chest, which capture over various periods of time the overall magnitude of the heart's electrical potential from twelve different polarisation angles, also known as "leads". The output of the sensing

TABLE I: Summary of EEG sensing equipment.

Producer	**Model**	**Grade**	**Mobility**	**Frequency range [Hz]**	**Channels**	**Sampling rate/channel [Hz]**	Reference
EGI	Geodesic 400	clinical	fixed	N/A	32-256 EEG	1000	[75]
Mindmedia	Nexus32	research	fixed	N/A	24 EEG	2048	[84]
Mitsar	EEG-202 series	research	fixed	0-150	24-31 EEG	2000	[85]
Nihon Kohden	Neuroxax 1200	research	fixed	0-1200	256 EEG, 16 DC	10000	[86]
Compumedics	GraelHD	research	fixed	0-3500	4 × 64 EEG	20000	[81]
Neurovirtual	BW series	clinical	fixed	0-500	25-50 EEG, 8 DC	2000	[87]
Nihon Kohden	Trackit	clinical	mobile	0-100	32 EEG, 4 DC	200	[80]
NeuroConn	Thera Prax	research	mobile	0-1200	22 EEG	32-4000	[88]
EGI	Geodesic 100	clinical	mobile	N/A	32 EEG	1000	[89]
Medicom MTD	encephalan EEGR	clinical	mobile	0.016-70	20 EEG	2000	[90]
Neurowerk	Neurowerk DB series	clinical	mobile	N/A	21-25 EEG	128-512	[91]
Compumedics	GraelHD	clinical	mobile	0-580	32 EEG	16000	[83]
Compumedics	Nuamps	clinical	mobile	0-260	40 EEG	1000	[92]
antNeuro	eego mylab	clinical	mobile	N/A	32-256 EEG	16000	[82]
Deymed	TruScan series	research	mobile	0.16-450	24-256 EEG	4000	[93]
Neurowerk	Neurowerk EEG mobile	clinical	wearable	N/A	16 EEG	128-512	[91]
antNeuro	eego sports	clinical	wearable	N/A	32-64 EEG	2000	[94]
Neurovirtual	BWmini	clinical	wearable	0-500	33 EEG, 2 DC	1000	[95]
NeuroSky	BrainWave	research	wearable	0-100	1 EEG	512	[96]
Mitsar	EEG-201 series	clinical	wearable	0.16-70	21 EEG	500	[85]
Emotiv	Epoc	research	wearable	0.2-45	14 EEG, 2 ref	128	[97], [98], [79]
Emotiv	Epoc+	research	wearable	0.16-43	14 EEG, 2 ref	256	[79]
Emotiv	Insight	hobby	wearable	N/A	5 EEG, 2 ref	128	[78]
Macrotellect	BrainLink	hobby	wearable	8-30	N/A	N/A	[99]
Interaxon	Muse	hobby	wearable	N/A	7 EEG	N/A	[100], [101]
Somnomedics	SOMNOsecren EEG 32	research	wearable	N/A	25 EEG	4-512	[102]
Neurosoft	Neuron Spectrum AM	clinical	wearable	N/A	21 EEG	1000	[103]
Deymed	TruScan wireless	research	wearable	0.01-1200	32-256 EEG	6000	[93]
Deymed	BFB	clinical	wearable	0.3-70	2 EEG	256	[104]

TABLE II: Summary of rest testing ECG equipment.

Producer	**Model**	**Grade**	**Mobility**	**Frequency range [Hz]**	**Channels**	**Sampling rate/channel [Hz]**	**Encoding [bits/sample]**	**Reference**
Cardioline	TouchECG	clinical	mobile	0.05-300	12	1000		[105]
Cardioline	AR600 series	clinical	mobile	0.05-150	12	500-1000		[106]
TSE	UCARD200	clinical	mobile	0.05-150	12	1600		[107]
Nasan	Simul-G	clinical	mobile	0.05-150	12	N/A		[108]
Longfian	ECG-1112	clinical	mobile	0.05-160	12	1000		[109]
Nasan	ESY-G1	clinical	mobile	0.05-150	1 (12 serial)	N/A		[110]
Nasan	ASAAN 1003	clinical	mobile	0.05-150	3 (12 serial)	N/A		[111]
Nasiff	CardioCard	clinical	mobile	0.05-150	12	250-1000		[112]
SpaceLab	CardioDirect 12	clinical	mobile	0-15	12	2000		[113]
CustoMed	CustoCardio 100 series	clinical	mobile	0-500	12	1000-4000		[114]
Kalamed	KES121 series	clinical	mobile	0.05-165	12	2000		[115]
Kalamed	KES1000	hobby	wearable	0.05-165	12	2000		[116]
Labtech	EC-12RM	hobby	wearable	0.05-150	12	512		[117]

is the electrocardiogram, which contains magnitude and direction of the heart's electrical depolarisation at any moment throughout the cardiac cycle.

In clinical contexts ECG sensing provides important information about the physical structure of the heart and the operation of its electrical conduction system (e.g. rate and rhythm of heartbeats, size and position of heart chambers, operation of implanted pacemakers etc.) and have been used extensively for diagnosis of various heart conditions (e.g. abnormal heart's muscle cells or conduction system, the effects of cardiac drugs). In addition, recent wearable ECG sensing systems have been extensively used for permanent monitoring of people with chronic cardiovascular diseases. For example, in [59] the author mentions a compact plaster sensor designed for wearable low cost cardiac healthcare, which uses a low power high resolution thoracic impedance variance for ECG monitoring. In Table II and Table III we summarise the popular ECG equipment available to date for resting and stress contexts, respectively, and list for each of them the essential sensing data-generator features.

- **Skin Conductivity:** Skin conductivity (SC) sensing is based on electrodermal activity (EDA), in which electric properties of the skin varies with environmental conditions and activity performed [127]. Human skin, especially that from the extremities of the limbs, e.g. fingers, palms, or soles of feet, display bio-electrical phenomena that are relevant to diagnosing a variety of psycho-physiological conditions [128]. These phenomena can be detected with sensors that capture changes in the electrical activity between two points on the skin over time [129]. Traditionally, electrodermal activity was associated

TABLE III: Summary of stress testing ECG equipment.

Producer	Model	Grade	Mobility	Frequency range [Hz]	Channels	Sampling rate/channel [Hz]	Encoding [bits/sample]	Reference
CustoMed	CustoCardio 200 series	clinical	mobile	0-500	12	1000-4000		[118]
Kalamed	KES121 series	clinical	mobile	0.05-165	12	2000		[115]
Eccosur	ECG-View	clinical	mobile	0.05-100	12	256-600		[119]
Thor	Thoriax PC	clinical	mobile	N/A	12	2000	12	[120]
SpaceLab	CardioDirect 12S	clinical	wearable	0-150	12	2000		[121]
Cardioline	CubeStress HD+	clinical	wearable	0.05-300	6+6 or 12, AVG	1000		[122]
Cardionics	CarTouch	clinical	wearable	N/A	12	1000	16	[123]
Labtech	EC12RS	clinical	wearable	0.05-150	12	1000		[124]
Amedtec	CardioPart 12 Blue	clinical	wearable	0-150	12	8000		[125]
Thor	Thoriax Home	hobby	wearable	0.5-40	1 (differential)	N/A	12	[126]

TABLE IV: Summary of skin conductivity equipment.

Producer	Model	Grade	Mobility	Conductivity range [S]	Channels	Sampling rate/channel [Hz]	Encoding [bits/sample]	Reference
Neulog	NUL-217	clinical	wearable	0-10 μS	1	100	16	[132]
Shimmersensing	Shimmer3 GSR+	clinical	wearable	0.2-100 μS	1	N/A	N/A	[133]
Mindmedia	SC Sensor	research	wearable	0.1-1000 μS	1	N/A	N/A	[134]
TMSI	GSR sensor	research	mobile	N/A	1	100	N/A	[135]
Mindfield	eSense	hobby	wearable	N/A	1	10	18	[136]
TMSI	GSR sensor	research	mobile	N/A	1	100	N/A	[135]

with the galvanic response of skin tissues (GSR - Galvanic Skin Response), in which skin resistance (impedance) varies with the state of sweat glands, when the glands are excited by the activity of sympathetic nervous system. Thus, galvanic skin response sensors were used to determine various aspects related to psychological or physiological arousal. However, recent insights showed that other sub-dermal tissues and physiological activity can also influence the electrical properties of the skin, in addition to the psychologically-induced gland activity. As a result, the concept of skin conductivity sensing has been enlarged to encompass the electric potential response (GSP - Galvanic Skin Potential), where sensors capture the voltage or changes in voltage measured between two electrodes conveniently located in two different locations on the skin.

Modern EDA sensing is largely used for capturing complex emotional and cognitive states [130], [128], in contexts like anxiety monitoring and control, stress detection, monitoring epileptic patients with drug resistance, or as part of polygraph equipment for lie detection in conjunction with other biofeedback sensors such as heart rate and blood pressure [131], or respiratory rate. Table IV provides a summary of the most popular skin conductivity monitoring equipment available to date, and list for each of them the essential sensing data-generator features.

- **Movement, positioning and gait sensors:** Recent advances in miniaturisation of inertial sensors led to an increase in monitoring of physical movement and positioning of humans in relation to different contexts. In medical sciences inertial sensors can measure and monitor neuro-degenerative conditions such as Parkinson's disease, patient's fall due to seizures or cardiac arrest, or patient's movement during physical rehabilitation programs. Another extensive use of inertial sensors is in sports and entertainment for monitoring and improving certain physical activities and the relevant cognitive-motor skills. Last, but not least, in recent years there has been a surge of studies which employed wearable inertial sensing for monitoring and investigation of social activity and behaviour.

  The various mobile, portable or wearable inertial sensors for human activity follow the same architecture and technology as the general inertial sensors we discussed earlier in the paper, consisting of combinations of accelerometers and gyroscopes for measuring acceleration along a sensitive axis and over a particular range of frequencies. Most of the sensing systems reported in the literature are based on piezoelectric, piezoresistive, variable capacitance, or MEMS transducers, which use variations of the classic inertial principle of operation: a core element (i.e. a mass) that responds to acceleration by proportional stretch or compression of another spring-like element. An thorough review on inertial sensors for human activity monitoring, with a focus on wearability can be found in [65].

  Table V summarises the most popular movement/activity trackers available to date, and list for each of them the essential elements of the fused activity sensing system.

2) Sensing machine agent activity: Surrounding technological systems are also one of the vital sources of data for autonomous systems. An autonomous unmanned road vehicle, for example, will gather data from various machine agents (i.e. peer-technology sources) that can be internal or external in relation to the vehicle. Internally, numerous sensors monitor the mechanical sub-systems involved in the normal operation of the vehicle, such as propulsion (e.g. crankshaft, camshaft, torque, oxygen, temperature sensors, etc.), breaking (ABS - Anti-lock Break System, EBD - Electronic Break Distribution), safety (impact, light, rain, proximity sensors), etc. Externally, numerous other sensors monitor the interaction of the vehicle with the environment, such as GPS for positioning, long and short range radar for obstacle detection, cameras for vision, etc. The

TABLE V: Summary of movement, positioning and gait sensing equipment. Legend: A - accelerometer, G - gyroscope

Producer	Model	Mobility	Format	Inertial sensors	Other bio-sensors	Reference
Angelsensor	Classic/M1	wearable	wristband	A, G	HR, skin temperature, oximeter	[137]
Apple	Watch	wearable	watch	A	HR	[138]
Basis	BasisPeak	wearable	watch	3 axis A	HR, GSR, skin temperature	[139]
Empatica	E series	wearable	wristband	3 axis A	HR, GSR, skin temperature	[140]
Empatica	E series	wearable	wristband	3 axis A, G	HR, GSR, skin temperature	[131]
Fitbit	Zip	wearable	pendant	3 axis A	-	[141]
Fitbit	One/Flex/Charge/Alta	wearable	wristband	unspecified motion sensor	HR, simplified ECG	[142]
Fitbit	Blaze/Surge	wearable	watch	unspecified motion sensor	HR, simplified ECG	[143]
Fitbug	Orb	wearable	watch	3 axis A	-	[144]
Garmin	VivoFit/VivoSmart series	wearable	wristband	unspecified motion sensor	-	[145]
Garmin	VivoActive series	wearable	watch	unspecified motion sensor	-	[145]
Google	Android-Wear	wearable	watch	market dependant	market dependant	[146]
iHealth	Edge	wearable	wristband	unspecified motion sensor	-	[147]
Jawbone	UP series	wearable	wristband	3 axis A	HR, GSR, respiration	[148]
LG	Lifeband Touch series	wearable	wristband	unspecified motion sensor	-	[149]
LG	watch series	wearable	watch	unspecified motion sensor	-	[150]
Medisana	ViFit series	wearable	wristband	unspecified motion sensor	-	[151]
Microsoft	Band	wearable	wristband	3 axis A, 3 axis G	HR, GSR, skin temperature	[152]
Misfit	Shine/Ray series	wearable	watch	3 axis A	-	[153]
Misfit	Flash	wearable	watch	3 axis A	-	[153]
Misfit	Ray	wearable	pendant	3 axis A	-	[153]
Misfit	Link	wearable	button	3 axis A	-	[153]
Nike	Fuelband	wearable	watch	3 axis A	-	[154]
Pebble	SmartWatch series	wearable	watch	3 axis A	-	[155]
Razer	Nabu/NabuX	wearable	wristband	3 axis A	-	[156]
Razer	Nabu Watch	wearable	watch	3 axis A	-	[157]
Salutron	LifeTrak series	wearable	watch	3 axis A	HR, ECG	[158]
Samsung	GearS series	wearable	watch	A, G	HR	[159]
Samsung	GearFit series	wearable	wristband	A, G	HR	[160]
Sensoria	SensoriaFitness	wearable	smart sock	3 axis A	3 pressure sensors	[161]
Sony	Smartband series	wearable	wristband	A	-	[162]
Sony	Smartwatch series	wearable	watch	3 axis A, G	-	[162]
Striiv	Fusion series	wearable	wristband/watch	3D accelerometer	-	[163]
Withings	Pulse O2	wearable	pendant	3 axis A	heart rate, oximeter	[164]
Withings	Go	wearable	pendant	3 axis A	-	[165]

situation becomes even more complex when speaking about the operation of an aircraft, or about the operation of a whole aircraft fleet, or an even more complex example, the operation of an autonomous air traffic control system, which actually handles an entire airspace. The resultant sensing mechanism is an overall equivalent sensor of very high complexity, which provides a fused global image of the data coming from the numerous internal and external sources.

In the previous section we discussed human activity sensing by considering the main categories of biophysical activity sensors separately. That discussion was pertinent because humans are identical from the perspective of the activity data they produce, thus autonomous systems need to embed a unique set of sensors in order to interact with different humans. However, in the case of sensing the activity of technical systems, there is a variety of sources the autonomous system may encounter during its operation. Thus, it becomes pertinent to discuss fusion sensing rather than the separate types of sensors. Sensor fusion [166] is the process of combining the sensory data coming from disparate sources in a way that (1) reduces the uncertainty resulting from separate treatment of the sources and/or (2) unveils properties of the sensed environment that were not obvious when using separate sources. Sensor fusion is receiving increasing attention from the research community, as the sensor technologies improve, together with the subsequent autonomous systems. Several thorough discussions can be found in the literature, proposing fusion methods with application in various fields of activity, such as control systems [167], thermal engineering [168], wearable robotics [169], or 3D computer vision [170].

Over time, numerous sensor fusion architectures have been proposed and categorised differently by different studies. From a project management perspective, in [166], the author describes abstract, generic and rigid architectures. The abstract ones constitute as ways to describe or explain the operation of a fusion system without guiding the engineers toward the actual implementation. Their importance lies in that they provide the initial understanding of the fusion problem. Generic architectures describe how to implement the system, but without specifying the technology (i.e. operating system, software and hardware, communication system). They are important for providing the engineers with complete know-how, while still leaving them the freedom to adapt their solutions to specific problems. Rigid architectures provide detailed design and implementation of the fusion system, guiding the engineers step-by-step in a strict manner. They are important when sensing systems are embedded in critical infrastructures or safety critical applications, where rework or re-implementations of the systems are not only unnecessary, but actually strongly avoided.

Another classification proposed in the literature [171], [172], [173] takes a conceptual approach and discusses four main categories - centralised, hierarchical, distributed (or autonomous), and decentralised - from which various hybrid architectures can be further generated [172]. This classification is perhaps the most popular for both researchers and practitioners, becoming in time a fundamental part of the undergraduate control engineering curriculum [171].

In centralised fusion architectures raw sensor data are transmitted to a central site, where they are aggregated to generate the information needed to produce a single fused description of the state/dynamics of the environment. In general, centralised fusion needs high processing power and data buses with high band-width in order to handle the high amount of raw data generated by sensors; thus, scaling up to complex systems may became a problem.

In hierarchical fusion, the lowest level processing elements transmit information upwards, through successive levels, where the information is combined and refined at each level, until it reaches the top level, where the full view of the state/dynamics of the environment is made available. Hierarchical fusion involves a certain processing capability on each level, and thus the load on the centralised (highest level) processor is reduced. In the same time the need of communication is reduced, since transmission of data/information is strictly controlled and limited to adjacent layers. However, while useful in general, this can be also a disadvantage due to the inability of data/information sources to communicate over more than one layer.Another major disadvantage of hierarchical fusion is that changes in a certain layer imply changes in all related sub-layers; this usually prevents the sensing system to be extended to incorporate more sensors.

Distributed fusion was meant to mitigate the drawbacks of centralised and hierarchical fusion, by providing autonomy to various sensing entities. In there architectures expert sensing agents with own processing capabilities make available to the system the output of their operation through a communication platform, often known as a “blackboard” from the architecture with the same name [172]. The obvious advantage of this approach is the distributed processing, with no central processing unit or layer. However, problems may arise due to the use of common communication/memory resource, i.e. the blackboard. Another disadvantage may be a potential loss in the accuracy of the global picture of the environment, due to the fact the picture is now dependant on simple agents with various processing capabilities.

Decentralised data fusion abandon completely all commonality aspects. A decentralised system consists of a network of sensor nodes, each with its own processing facility, and no central or common communication facility. In this case, fusion happens in each node, locally, by the sensor unit using local observations and information communicated from neighbouring nodes. Advantages of the decentralised approach are high scalability and modularity, where dynamic changes of the sensor number and structure are easily supported. Thus, the resultant sensing systems are highly resilient to loss of nodes. However, due to the lack of global communication or knowledge, the accuracy of the fused picture of the environment is highly dependant on the network features, i.e. topological features like centralities, entropy and other statistical metrics.

Numerous architectures have been proposed over time, from as early as 1960s to present days, for each of the four (plus the hybrid approaches). The interested reader can find thorough reviews of the general fusion architectures in a number of studies from different historical periods [174], [166], [172], [173]. However, it is outside of the scope of this study to provide insights in the individual architectures, since our purpose was to bring into attention the complexity that a sensor-based autonomous system must handle, in terms of the amount and variety of sensory data coming from the surrounding technological systems. Thus, we emphasize once more on the idea that big sensory data can be in the same time an enabling and a disabling factor in the operation of the autonomous systems.

C. Computers

Once data-driven and model-driven information become available, processing starts. Be it in the form of signal processing, feature selection, clustering, classification, or any form of processing that is needed to make a judgement, the machine needs to rely on its processing abilities to process these information. Processing today extends beyond the classic Von Neuman computer architecture to other forms of processing.

In the most general understanding, a computer is an entity that conveniently alters a set of inputs to generate a set of desired outputs. Since digital computers started to be used on large scale, and became indispensable to modern life, the general perception of a computer narrowed to its digital incarnation, where inputs are reduced to binary data and alteration towards the desired output is made through binary calculation. However, computing was and is not limited to digital manipulation of binary input data, since numerous entities exist that can perform the computing analogically if talking about machine entities (e.g. mechanical, electrical, optical, electronic) or biologically if talking about natural entities, such as the neural sensory processing in animals with a central nervous system [175], [176]. Given that in this study we discuss autonomous systems considering the big sensory data they have to process during their normal operation, it is natural to bring into attention mostly the digital computing machines available to date, with a focus on their ability to deal with these data. However, we consider useful to discuss the other two categories, analog computing and neural-inspired computing, in order to offer a more comprehensive and integrated view on the ways that are open to autonomous systems.

1) Analog computing: Analog computers use the continuously changeable aspects of a physical phenomenon of interest to alter the inputs towards the desirable output. Virtually, any physical process that models some computation can be seen as an analog computer, including finding the shortest path between two points by stretching a string between them. They have been heavily used in scientific and industrial applications in classic automation and control systems of the pre-digital era, when digital computers of the time lacked sufficient performance. However, they have been less and less used, becoming eventually obsolete as the digital computers came into place and became more powerful and versatile. There are still domains where they are still in use in some specific applications, where classic automation is required due to operation costs and environmental conditions. Extensive historical views on analog computers can be found in [177], and also in a special issue of IEEE Control Systems dedicated to the history of analog computers, from which we cite here the editorial article [178] signed by Lundberg.

Examples of analog computers can go from the ancient nomographs and sextants, to slide rules, and further to complex contemporary automation and control systems such as navigation systems, analog auto-pilot systems for vehicle navigation, military weapon system controllers, analog signal processors, etc. In essence, virtually all industrial process control systems in the pre-digital era used analog computers, known at the time as controllers, to automatically regulate temperature, flow, pressure, or other process variables and parameters. The technology of these controllers ranged from purely mechanical systems to emulation of physical phenomena through electronic analog components, and later through analog integrated circuits.

The strongest point the analog computers make when compared to their digital counterparts is the computation speed; i.e. the analog computers operate in real-time once they have been set up. The computation in analog computers is not explicit like in digital computers, but rather implicit, inherently contained in the materials or phenomena they use as computation mean; thus the delivery of the output based on the input is virtually instantaneous. For example, an integrator can be built as an electric circuit by conveniently placing a resistor and a capacitor in a feedback loop, so that the output delivers a steadily changing voltage for a constant input voltage.

A major disadvantage is that analog computers are designed for performing calculation in particular contexts outside of which they cannot function properly. Their setup involves choosing (1) the scale and limits for the inputs and outputs given the material or phenomena used for computation, (2) the set of pre-operational (initial) and operational conditions, and (3) the mechanism or physical structure (i.e. interconnection of computing elements) of the computation in order to assure proper solving of the given problem. When launched into operation variables cannot exceed computer's predefined operational conditions. For example, if the above mentioned integrator is to be used in humidity and temperature conditions that exceed the limits established during design stage based on the material/electrical tolerances indicated by manufacturer for the component elements (resistor, capacitor etc.), then, unacceptable calculation errors may occur. Also, if the integrator is to be used as a differentiator, this can be done only by changing the physical arrangement of the resistor and capacitor in the feedback loop, together with their values. In other words, a different circuit - that is, a different analog computer - is needed.

Analog computers can be found or imagined virtually anywhere and in anything in nature, however, the literature mostly speaks about the mechanical and electronic versions, due to the fact they account for the majority of the modern time achievements in both industry and research in pre-digital era. Both categories stand out due to their theoretical importance at the time or due to the popularity they gained in practical industrial applications. In the following we briefly discuss them in order to create a complete image of the analog computing field.

- **Mechanical analog computers:** Mechanical analog computers used over the years were able to implement the usual arithmetic, trigonometric, and geometric operations, and also complex analytic and algebraic transformations or arbitrary functions by conveniently combining various mechanical elements, such as rotating shafts, discs and gears, racks and pinions, cables and pulleys, springs, differential mechanisms, etc. For example, addition and subtraction have been implemented using precision bevel-gear differentials. Integration was implemented with a rotating disk (integrator variable) and a pickoff wheel positioned on the disc at a radius proportional to the integrated variable. Also, coordinate conversion from polar to rectangular was implemented with two coaxial discs and a sliding block with a pin on it. While these examples are important for analog computers, they are certainly not exhaustive. Extensive reviews of these computers can be found in the literature of the 1960s and 1970s [179], and more recently in [176].
- **Electronic analog computers:** Electronic analog computers emerged as a result of the similarities between the equation-based mathematical models of mechanical components, and those of the electrical components. Complex and heavyweight mechanical computational devices were gradually replaced with their electrical equivalents which could be constructed with operational amplifiers and other linear (resistors, capacitors, inductors) and non-linear (diodes, transistors) passive components. The resultant electronic analog computers were less expensive to construct, safer, and easier to modify for new problems in comparison with the mechanical counterparts. Gradually, more and more technological and natural phenomena could be represented through analogy by using electronic circuits, which could easily perform a wide variety of simulations. For example, an application would be to integrate a signal representing water flow, producing an output signal representing total quantity of water that has passed through a flow-meter. Voltage can be the analogy of water pressure and electric current the analogy of flow-rate in terms of volume/second. Then, the integrator can provide the total accumulated volume of liquid, using an input current proportional to the flow rate.

  Electronic analog computers contain may contain, depending on their purpose, tens, hundreds or thousands of operational amplifiers combined with relevant passive components in various feedback schemes, in order to implement mathematical operations and functions. The core operations and functions are: addition, subtraction and comparison; differentiation and integration; multiplication, division and inversion; exponentiation and logarithm. More complex arbitrary non-linear functions have been also implemented, however, they had limited precision (typically three to four digits) and were difficult to implement, requiring laborious and complicated feedback loops with combinations of linear and non-linear components in the feedback loops of operational amplifiers. An in-depth discussion on electronic analog computers can be found in [180].

The major disadvantages of the electrical analogy are (1) the limited dynamic range, i.e. electronic components can represent limited variation ranges of the variables they emulate (compared for example with the virtually infinite dynamic range of the floating-point representation used by digital computers), and (2) the noise levels introduced by the electronic components and circuits. In general, electronic analog computers are limited by non-ideal effects induced by the undesired deviations of the core variables of the electric signals: DC and AC amplitudes, frequency, and phase due to material and environmental limitations [181]. The limits of range on these characteristics limit analog computers. Some of these limits include the operational amplifier offset, finite gain, and frequency response, noise floor, non-linearities, temperature coefficient, and parasitic effects within semiconductor devices.

- **Revival of analog computing:** Analog computation certainly went into an eclipse in the last few decades, due to the rise of digital computers, however recently more and more authors acknowledge that analog computation is inherent to most of the existing natural systems [182]. In [175] the author even note that some digital computing paradigms are actually unsuited and unnatural (pun intended) to natural computation. Thus, recently new iterations of analog computers have been proposed, especially in their electronic version.

  Recent development of semiconductors and very large scale integration went mostly in the direction of integrating digital circuits. However, recently, several analog VLSI circuits have been proposed, implementing energy efficient and instantaneous complex mathematical calculation. Cowan and colleagues [183] proposed a single-chip VLSI analog computer with 80 integrators and 336 other programmable linear and nonlinear circuits manufactured in $250nm$ CMOS technology. The chip was intended for use as an analog mathematical coprocessor for general purpose digital processors, and was rated by authors with a consumption of only $300mW$. Later, Schell and Tsividis [184] proposed a hybrid signal processor where fixed sampling and clock rates of a conventional DSP were eliminated to create a clock-less digital signal processor. The processing unit was manufactured in $90nm$ CMOS technology and operated in continuous time with a power source of 1 Volt, offering no aliasing, and very fast transient performance in power electronics due to lack of sampling. Recently Guo and colleagues [185] demonstrated a continuous-time hybrid computing unit in $65nm$ CMOS technology, capable of solving nonlinear differential equations up to 4th order, and scalable to higher orders. The authors rate the energy efficiency of the processor at $12mW$, and demonstrate the use of the unit in a low-power cyber-physical systems application, where the problem is solved in $0.84\mu s$.

2) Digital computing: In general, analog computers, analog signals or other analog processes implicitly involve an *analogy* or a systematic relation with the physical processes they model or simulate. They have inherent advantages, as discussed in previous section, such as speed and low energy consumption, given by the fact they operate on continuous representations of real processes by using continuous processes *analog* to real ones. However, the *analogy* itself also creates one of the most important drawbacks, that is the lack of flexibility, since a signal that was found to be *analog* to a physical process typically cannot be used with ease for another process. Digital computers, on the opposite, operate on discrete representations of real processes by means of discrete steps. Thus, real processes are represented by strings of binary symbols that do not have an explicit or direct physical relationship to the modelled processes. In [176] the author notes that in a digital computer the relationship, i.e. *analogy*, is more abstract than simple proportionality. This extra abstraction steps made the digital computers extremely versatile, in that the discrete modelling and operation allow their use in a large variety of applications, virtually without any hardware change. With this important advantage, doubled by a rapid increase in speed and energy efficiency, the digital computing eclipsed the analog ones, despite a number of disadvantages. Some of these advantages, which we mention here for completeness reasons, are as follows: (1) sampled signals used in digital computing suffer from aliasing effect, which leads to power dissipation on superior harmonics of the digitised signals and to the need of precise filtering; (2) digital signals may present important undesired behaviour in the transient domain; (3) discrete representation can lead to latency when data are transferred between various internal components of the processing architecture.

Today, digital computers reached an unprecedented diversity, with their performance measured with respect to certain attributes that make them fit-for-purpose, rather than a unique performance metric. In general, digital computing is subject to a trade-off in which multiple aspects are involved, such as processing-related features (e.g. energy consumption, memory capacity, throughout, latency, etc.), operational features (e.g. size, weight, reliability, expandability, etc.) on top of which there are of course the financial costs of acquisition and operation. Thus, the overall performance of a digital computer can be measured using numerous metrics depending on its purpose. For example, a computer can be CPU bound - heavily used for raw computation, input-output (I/O) bound - mainly used as a server, or memory bound - mainly used for memory intensive tasks such as video editing. From the point of view of autonomous systems though, two aspects are essential: raw processing power and energy efficiency. We discussed earlier in the sensors section how portability and miniaturisation are the engines that propelled the sensors field to nowadays advances, which are unprecedented in the history of human kind. Thus, the autonomous systems making use of sensors must fall in the same philosophy of fast processing of data versus high portability.

Processing power was first associated with processor's clock rate, that is, the frequency (number of cycles per second) of the main clock of the CPU. However, this turned misleading as digital computing evolved and processors became able to execute more and more instructions at every processor cycle. Thus, a higher clocked processor may have a lower throughput than a lower clocked processor if the latter can execute significantly more instructions per cycle. Thus, another measure of the processing speed was the number of instructions per cycle, which in conjunction with processor's clock rate could give a better estimate of the speed with at which a certain task can be processed. However, this was also subject to ambiguities since different processor architectures can embed different sets of instructions, and/or can implement similar instructions differently. Perhaps the most accepted speed metric for modern processors is related to the amount of work done per unit time, i.e. the throughout, and is measured as the number of floating point operations per second (FLOPS).

Energy efficiency is the other vital aspect in the operation of autonomous systems, mostly for portability reasons, especially because the usual perception of an autonomous system involves embodiment and mobility. Disembodied autonomous systems can be brought into discussion, like autonomous decision-making systems hosted on mainframes of fixed computing facilities, in which case the energy consumption may be important from a cost perspective. However, in a sensing-based autonomous system, mobile embodiment is the pertinent form of existence, which leads to high importance of the energy efficiency offered by the processing units.

- **Classic processing-storage digital computing:** Computers in this category generally refer to architectures in which memory is conceptually separated from the processing unit, and data to be processed are permanently exchanged between the two. In general, the processing-storage architectures describe digital computers consisting of a processing unit, memory units that store programs (instructions) and data, input-output mechanisms, and external mass storage. Several architectures have been proposed over time, which differ through the way the exchange takes place within the component elements of the architecture. Of these, *von Neumann*, *Harvard* and *modified-Harvard* architectures are those that provided the majority of the digital computers available to date [186].

  The von Neumann architecture proposed a design in which both data and instructions are stored in a unique memory, and thus they share one data bus for communication with the processing unit. The architecture was simple, through the use of only one data bus and one memory, but also introduced a major disadvantage, known as *von Neumann bottleneck*. The bottleneck consisted in the impossibility to fetch instructions and operate on data in the same time due to the shared bus that could only handle one type of memory items at a time. This led to limitations of the overall throughput (processing power), especially in data intensive applications, due to the fact the CPU was continually forced to wait for data to be transferred to or from memory.

  Several methods to overcome this bottleneck have been proposed over time, and included sole or combine use of: a cache between CPU and memory, separate caches for data and instructions between CPU and memory, separate access paths (bus) for data and instructions, on-chip single- or multi-layer CPU cache, pipeline mechanisms with prediction algorithms for data/instruction transfer, etc.

  The Harvard architecture was one of the most successful improvements proposed for overcoming von Neumann bottleneck. The architecture introduces a design with physically separate storage and transfer for instructions and data, thus the design features dedicated data memory and instruction memory, as well as dedicated data bus and instruction bus. Thus, in computers build on Harvard architecture, the CPU can read an instruction and operate a data transfer at the same time, leading to higher processing speed without using memory cache. In addition, since data and instructions are completely separated (storage and transfer path) their features and implementation can be different, such as different word size, different timing, different implementation technology, or different memory address structure, as well as different data transfer standards.

  The main use of Harvard design, is in applications where the energy saving and implementation costs, so that processor cache must be avoided, while still keeping a high processing speed. In signal processing, the digital signal processors (DSP) generally execute small, highly optimised processing algorithms, hence their memory usage is well determined, so that memory cache is not needed. In the same time, DSP applications need high parallelism, so multiple address spaces must be used, that usually need to be addressed in parallel. Thus, some DSPs feature multiple data memories in distinct address spaces to facilitate paralel processing approaches like Very-Long-Instruction-Word (VLIW) and Single-Instruction-Multiple-Data (SIMD). An example is the Texas Instruments TMS320 C55x processor family, which includes five parallel data buses two for writing and three for reading, and one instruction bus. Micro-controllers are another category of computing units implemented in Harvard architecture. They are typically used in embedded systems, where the applications need small amounts of program (instructions) and data memory, but need high speed processing with low energy consumption. Thus, designs with concurrent instruction and data access and no cache are preferred by most manufacturers, such as Atmel's AVR family or Microchip's PIC family.

  However, while providing high processing speed and low energy consumption for a range of applications like RISC or embedded systems, the architecture was not appropriate for full-sized modern processors used in general-purpose computing, such as personal computers and laptops, and more recently tablets and smart-phones. This happened especially due to the unbalanced development of various elements of the computing architecture, where for example the speed of CPU, memory, bus and external storage experienced very different growth pace.

  Thus, various improvements of Harvard architecture have been proposed, all known under the umbrella of modified Harvard design. In essence the modified versions are still Harvard architectures through that the CPU is able to access data and instructions separately, but in the same time they resemble von Neumann design by storing both in a unique memory. The use of *split-cache* system is the most common modification, used in most modern general-use processor families from INTEL, AMD, ARM, Motorola, IBM, SUN, etc. In this design, separate instruction and data caches are used by CPU to access a common address space. When the CPU executes from caches the computer falls onto the standard Harvard design, and when it accesses the common memory it falls onto the von Neumann design.

- **Highly parallel processing on GPUs:** A significant step towards increasing processing speed was made during the mid 2000s, when general-purpose computing tasks started to be ported from CPUs onto graphic processing units (GPU) [187], [188]. GPUs are specialised processing units designed to accelerate the creation of images in a frame buffer (that is further output-ed to a display) through a multi-core structure that is able to process large blocks of visual data in parallel. GPUs operate in general at lower clock frequencies compared to CPUs, but embed very high numbers of simple cores working in parallel, overpassing overall the capabilities of even the fastest multi-core CPUs available today. However, the limitation of GPUs was that their simple cores were designed for handling graphics data only. Early GPUs embedded specialised cores were used for texture mapping and polygon rendering, with later additions that included various geometric calculations, e.g. rotation/translation of vertices into different coordinate systems. More recent GPUs added programmable shaders which for manipulating vertices and textures, oversampling and interpolation techniques for anti-aliasing, and very high-precision color spaces. Today, typical specialised cores in modern GPUs are pixel shaders and vertex shaders (recently combined in "unified shader architecture"), texture mapping units, and render output units.

  Since most graphical computations involve vector and matrix manipulation, researchers and practitioners have increasingly studied the use of GPUs for non-graphical calculations. Since the graphic cores were unable to process general-use data, the usual CPU applications had to be transformed into graphical forms in order to be processed by GPUs. Thus, early attempts to perform general-purpose computing on GPUs focused on reformulating at conceptual level the general computation problems in terms of graphics primitives, in order to be compatible with OpenGL or DirectX [189], the two major APIs available at the time. This difficult translation was later abandoned with the emergence of the GPU computing ecosystems (API + SDK), such as the proprietary CUDA from nVidia [190], which was the first API to allow the transfer of C, C++ and Fortran code (and later Matlab, Python and C#) straight to GPU, with no assembly language required. Later, in 2007, AMD-ATI released its own proprietary ecosystem Stream, which followed the ATI developed Close-To-Metal and further evolved into the recent Accelerated Parallel Processing (APP) ecosystem that offers support for newest AMD's Heterogeneous System Architecture, in which CPU and GPU cores are designed to work together in a single accelerated processing unit (APU) [191]. Recently, hardware-independent platforms have been released, such as the proprietary DirectCompute from Microsoft as part of DirectX API [189], and the open-source OpenCL [192]. Through these, the general-purpose applications can be run through and benefit of GPU speed without requiring conversion of the data into graphic form.

  CUDA and Stream/APP ecosystems remained bond to nVidia and AMD GPUs, respectively. nVidia GPUs starting with GeForce 8 series were by default hardware enabled for CUDA, while AMD's GPUs starting with Radeon500 family were by default hardware enabled for Stream (and later for APP). Recently, the open source platform-independent OpenCL, developed by Khronos Group [192] became popular for that it allows development of code for both GPUs and CPUs, and received support from all major chip manufacturers, Intel, AMD, Nvidia, and ARM, whose most recent chips are OpenCL-enabled.

- **Reconfigurable processing:** Apart from the processing power and energy consumption, another aspect becomes more and more important in the context of sensor-enabled autonomous systems. Since the nature of these systems, as we explained in earlier in the paper, is to operate in dynamic environments, it becomes important to be able to adapt not only the decision making, but also the processing itself to the task of interest. However, while this is possible on general use computers through software, it is not possible at lower hardware level. Thus, an increased interest have been shown for the use of FPGAs (Field-Programable Gate Array), which by design have the potential to be reprogrammed at runtime, being in essence computing units that can reconfigure themselves to suit the task of interest [193]. Typically, FPGAs contain arrays of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be connected conveniently to perform a variety of operations, from simple logical operations like XOR, to complex combinatorial functions, mathematical functions or even emulate whole processors. Historically, they have been used as "breadboards" for low cost prototyping, in order to avoid the high non-recurring engineering costs of ASIC (Application-Specific Integrated Circuit) development. Thus, chip manufacturers could develop their hardware on standard FPGAs, and then manufacture at large scale their final version as an ASIC that could not be modified once the final commitment to a design has been done. For the same reason, FPGAs were also preferred for designing and manufacturing of chips, in applications where the volume of production was small and the development of ASICS was not worthy at all. Initial FPGAs were bond to these purposes also because of their low performance, combined with large physical dimension and high energy consumption [194]. However, with the recent advances in semiconductor technology, transistor gates and memories, the latest FPGAs can rival not only with their ASIC counterparts, but also with full-sized processors in multiple respects, such as speed, dimension, energy efficiency and cost, while having the advantage of reconfigurability. Recent FPGAs began to take over larger and larger functions to the point where they are even able to implement so-called "soft processors" that emulate the operation of powerful computers, either in ASIC domain or in general-computing domain. FPGA-based soft processors can be seen in various reconfigurable System-On-Chip (SoC) solutions, which incorporate in the FPGA fabric a variety of elements, such as ARM cores, networking modules, radio modules, DAC/ADC converters, and other peripherals. Examples of these FPGA are those from the major chip manufacturers, such as Microsemi SmartFusion series, Xilinx Zynq series, or Altera Arria series. A detailed review of the high speed FPGA SoC and soft processors can be found in [195].

Due to these technological advances, FPGAs are used today in complex applications that include digital signal processing, software-defined radio, medical imaging, computer vision, speech recognition, cryptography, bio-informatics, computer hardware emulation, radio astronomy, metal detection and a growing range of other areas. Another recent use of FPGAs is hardware acceleration, where FPGAs are used as coprocessors, to assist a general-use processors in acceleration of certain parts a task or algorithm [196]. Recent use of FPGA coprocessors was announced by Microsoft and Google, who started to use FPGAs to accelerate high-performance, computationally intensive data centres that operate their search engines, i.e. Bing and Google respectively.

3) Neuromorphic computing: Neuromorphic engineering, also known as neuromorphic computing, has its roots in the analogic nature of natural biological computation and is based on the transfer of the morphology of individual into electronic circuits. In essence, the resultant computing architectures and implementations are analog computers which contain circuits that mimic the electrical activity of neuro-biological nervous system, that is, described by the spiking neuron models. From the point of view of a sensor-based autonomous system, neuro-morphic computing seems like the obvious way to go, since the sensor-based operation is the essence of life, and thus any existing digital artificial sensor-based system, no matter how fast or sophisticated it is, is still far from what biological sensory processing can offer.

Most of the efforts in neuromorphic computing concentrated on the formal spiking models, with some notable instantiations of the theoretical models proposed in the last decade.

In 2006, researchers at Georgia Institute of Technology published the details of a field programmable neural array [197]. The chip was the first in a line of many increasingly complex arrays of floating gate transistors that allowed programmability of charge on the gates of MOSFETs to model the channel-ion characteristics of neurons in the brain and was one of the first cases of a silicon programmable array of neurons. The authors describe the chip as an analog circuit capable of accurately emulating large complex cells, or multiple less complex ones. They note its analogy to an FPGA in which the binary transistor gates were replaced by biologically relevant circuit components including active channels, dendrites, and synapses. A routing capability was added to these elements in order to transfer outputs from cells to individual synapses.

In 2012, Spintronic Rand Purdue University [198] announced a neuromorphic chip that used lateral spin valves and memristors. Their design was a hybrid structure in which a programming current passing through heavy metal generates spin-orbit torque to modulate the spintronic device conductance, creating a nanoelectronic synapse. The spintronic synapse was then interfaced with CMOS neurons operating in transistor subthreshold region in order to form a network of spiking neurons. The authors claim that the architecture can emulate the operation of biological neurons and reproduce various brain-like processing abilities with significantly lower energy consumption than conventional digital chips.

SpiNNaker [199], proposed by the neuromorphic engineering research group at Manchester university, is a massively parallel, low power, neuromorphic computing board designed to model very large, biologically realistic, spiking neural networks in real time. The board consists of 65,536 identical 18-core processors, meaning 1,179,648 cores in total. Each processor has an on-board router to form links with its neighbours, as well as its own 128 MB of memory to hold synaptic weights. Each core is an ARM968 chip manufactured using a 130 nm process. The resultant chip is not a purely hardware neuromorphic computer, but rather a massively parallel digital computer, with over one million cores, which can simulate one thousand neurons per core, leading to a total of one billion simulated neurons.

In 2014 Stanford University announced Neurogrid [200] structure. Neurogrid is a 6.5x7.5 square inch circuit board composed of 16 custom designed chips, referred to as NeuroCores. Each NeuroCore's analog circuitry covers 12x14 square millimetres and is designed to emulate 65536 neurons, thus the board can model a total of over one million neurons, and 6 billion synapses implemented in a binary tree with links, with operates at a total of 80 million spikes per second. The authors claim a consumption of only 5 watt at full processing load.

Perhaps the most advanced neuromorphic chip is TrueNorth proposed by IBM [201]. TrueNorth is built in CMOS technology and consists of 4096 hardware cores, each of them embedding 256 programmable hardware silicon neurons, which in turn have 256 programmable synapses. Over all the chip embeds over a million neurons, with a total of over 268 million synapses, on a physical CMOS wafer containing 5.4 billion transistors in 28 nm technology. The authors claim an energy consumption of less than 100mW/chip, meaning less then 20mW/cm^2, or 26pJ per synaptic event. They also claim that the chip implements the first fully event driven digital mixed synchronous-asynchronous neuromorphic architecture. Applications of the chip in real tasks showed a consumption of 70mW while a recurrent network at biological real-time, and 65mW in a multi-object detection and classification application, with 240x400-pixel 3-color video input at 30 frames-per-second. TrueNorth is claimed to be the first neuromorphic implementation that distances entirely from the classic processing-storage paradigm, through the fact the neurons are silicon based not emulated by conventional gates. Thus, the classic thinking of how to program a computer also cannot be applied any more, since concepts like memory access, data buses or CPU do not exist in TrueNorth design.

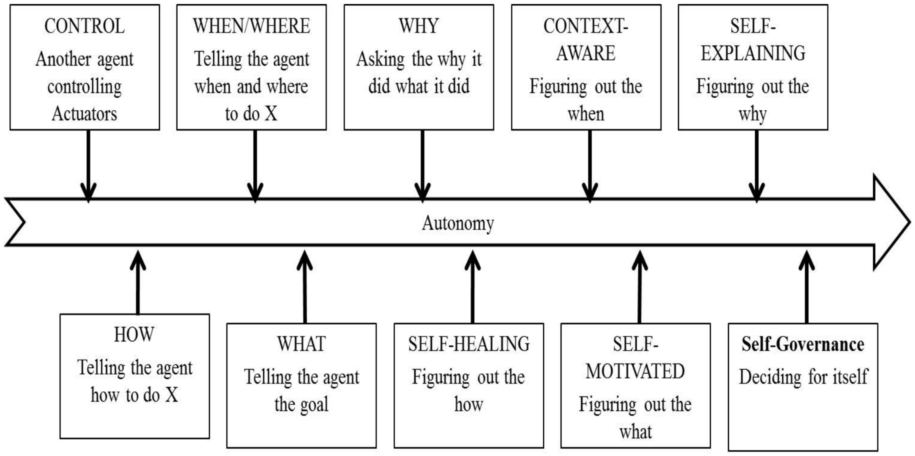

Fig. 5: Autonomy Timeline.

The authors designed for this reason an end-to-end ecosystem complete with a custom simulator and a new programming language for composing networks of neurosynaptic cores, called corelet [202] implemented in Matlab code. With this, IBM also offers an integrated programming environment with corelet libraries, conventional algorithms and applications ported into the neuromorphic paradigm, and a teaching curriculum called "SyNAPSE University".

D. Human-Machine Interface Big Data Problems

In our discussion of the sensor technologies to connect humans to machines, we avoided some of the classic and data intensive sensors such as different types of cameras and microphones. The Psychophysiological sensors reviewed are not too data intensive. The maximum EEG sampling rate was 20kHz. With a maximum of 256 sensors, we can have a 5mHz information flow from the human scalp. ECG and other sensors are not worth discussing from a big data perspective. Thus, it seems that the three most demanding data sources are imagines, EEG and sound in order of the complexity of data that can be acquired through these channels.

To process these data, clearly neuromorphic chips can offer a leap forward in the ability of artificial AS to perform real-time complex sensing and processing tasks. However, it is still too early stage for this technology and is very unlikely that the technology will be available in large scale and with a general-consumer accessible strategy in the near future. As such, we are back to the von Neuman and Harvard architectures, as the most feasible processing to be used today. This also includes GPUs and reconfigurable chips such as FPGAs.

From the 6 different Vs defining a big data problem, we comment in what follows on how the human-machine interface problem is a big data problem. The issue of volume arises specifically when dealing with video, EEG and voice. However, the main big data challenges within the context of this paper are more centred on the other Vs, which are all apparent in this context. The timeseries nature of most of these data, along with the chaotic nature of human mental processing, makes the problem to be susceptible to high velocity and high veracity. The multi-modality required to triangulate different sources of data to understand human mental states make the problem susceptible to high variety and variability features. The use of these data to make informed trusting decisions and facilitate human-machine interaction in TAS may depend on the application and context, but in general we expect it to have clear high value.

In summary, TAS bring big data challenges that need to be considered when designing these systems.

IV. Trusted Autonomous Systems: The Big Data Challenge Revealed

This section concludes the paper with the big data open challenges for trusted autonomy. In Sub-section IV-A, we will discuss autonomy to contextualise the discussion for the reader. This is followed by bringing together the discussion on Trust and the discussion on autonomy under the common objective of this paper: trust-aware autonomous systems in Sub-section IV-B. Sub-section IV-C offers some insight into some open challenges and future research directions.

A. The Road to Autonomy

The evolution of autonomy in Figure 5 follows similar principles to the evolution of trust. Autonomy is not a binary concept. The concept exists on a wide scale with complete absence of autonomy at the one end and full autonomy at the other.

We will explain autonomy using the WH-clauses: How (strategy), When/Where/Whom (context), What (goal), and Why (rationality). These questions are incremental in nature. Without the ability to develop strategies, an actor can't manage and/or survive a context. Without the ability to manage contexts, the actor can't deal with goals. Without the ability to manage goals, the actor can't explain and/or reason about its own behavior. The abilities to develop strategy, manage context, manage goals, and explain and reason can be hard-wired within an actor. When the actor is able to develop these abilities by itself, it develops 'self' and it rises up in the hierarchy of autonomy. This ability to develop 'self' does not happen in vacuum. It is another set of algorithms that work on a higher level of abstraction. Similar to learning to learn, we still need to use learning techniques, but with an appropriate representation, we can develop learning techniques that are able to learn on their own.

The left hand side of the diagram starts with a zero-autonomy; a dictatorship master-slave relationship, where zero autonomy is equivalent to a slave with full obedience to its master. An example we will use here to demonstrate the concept is in the case of a bicycle. As a machine, it does not do anything unless an external force (human or environment) exerts a force on it.

Let us add an electrical motor to the bicycle. The human may press a button and it will start to move. The motor offers a control without being aware of context. It will move in the same manner independent of whether it is night or day, moving on a road or in mud, and whether it is operated by an adult or a child. Once we add sensors and increase the level of intelligence in the controller, the machine starts to be 'context-sensitive'. It will behave differently in different contexts. We will follow a similar philosophy to Abowd et.al. [203] to differentiate between sensitivity to context; that is, the machine is using context information, and what we will discuss later on as 'context aware'. For now, this machine is not 'context aware' yet.

We can continue to improve the automation of the machine by adding a planner and a GPS system to offer it an opportunity to decide on goals. The human may ask the machine to go to a restaurant and the machine will choose the restaurant, plan the route, and take the human to a restaurant of its choice. The human may then need to ask the 'why' question; that is, why a certain route was chosen. In this case, we need to use the planner within the machine or add another layer of reasoning to offer the human an explanation of the choices made by the machine.

While we do have a smart machine that automatically decides on a route, adapts its control strategy to the information it is sensing from the environment on terrain and driver characteristics, is able to plan automatically, and reason about its action, the level of autonomy of this machine is still very limiting. What we have up to now is smart automation, with little autonomy. Taking this level of intelligence to autonomy is the vision of 'autonomic computing' that was first introduced by Paul Horn [204] in his 2001 keynote address to the National Academy of Engineers at Harvard University.

The ultimate aim of autonomic computing, or what is known now as 'self-X', is to have a self-governed system. This created a list of characteristics to make this dream a reality including: the system needs to have its own identity, be able to self-configure, self-optimize, self-heal, self-protect, and become self-aware. We will depart from some of these characteristics as we see the evolutionary steps we will discuss for autonomy to subsume some of them.

Autonomy starts to increase incrementally as we add more self-x capabilities with the ultimate goal to achieve total self-governing. The first feature is 'self healing', the ability to recognize problems (self-diagnosis) and recover automatically. The system needs to be able to be equipped with the mechanisms for it to discover its own strategies towards maintaining its health level and functionality at the right level.

Moving in the hierarchy from context-sensitive to context aware, the system needs to extend its ability to sense the environment surrounding it to an ability where it senses the user's tasks, user's intents, and know how to contribute to the optimization of this task. This definition is consistent with Abowd et.al. [203] who refer to a system as context-aware "if it uses context to provide relevant information and/or services to the user, where relevancy depends on the users task."

A context aware system with the ability to do self-healing is already too advanced in the autonomy hierarchy but still unable to exercise full autonomy. It needs to have the ability to autonomously generate its own goals through self-motivation and to reason in its own way through self-reasoning. These features allow the system to decide for itself what it needs to do and develop its own rationale for the how, when, where and what.

With the above self-X capabilities, a system becomes fully autonomous when it is able to integrate across these capabilities to form a self-governed whole. Here, autonomy reaches its maximum horizon.

B. Trust-Aware Autonomous Systems

The road to trusted autonomy is not a simple addition of trust to autonomy.

A trusted autonomous system (TAS) does not operate in a vacuum. It interacts with other TASs and within an overall context. It needs to take into account the complex interaction of the two concepts, trust and autonomy, and the multi-actor environment they operate within.

Each actor has its own objectives, goals, behavioral attributes, capabilities, and resources. While these actors individually may be trusted, the system as a whole may not be. The simple example is counterintuitive and can better be explained with the statement: if everyone does the right thing, it does not mean that the group is doing the right thing. The challenge in this statement lies within the boundary of the word 'right': right for whom and from whose perspective, and who has the authority to define what is right? What is right from an individual actor perspective may not be what is right from a system perspective. An actor may wish to operate optimally (e.g. a robot would like to reduce energy consumption to extend battery life), but the system may need the task to be completed as quickly as possible, which may result in a non-efficient use of the robot's battery. These sorts of conflicting objectives between the individuals and the group, the groups and the supergroups, and the supergroups and the society create a level of interdependency that complicates the concept of trust in social dynamics.

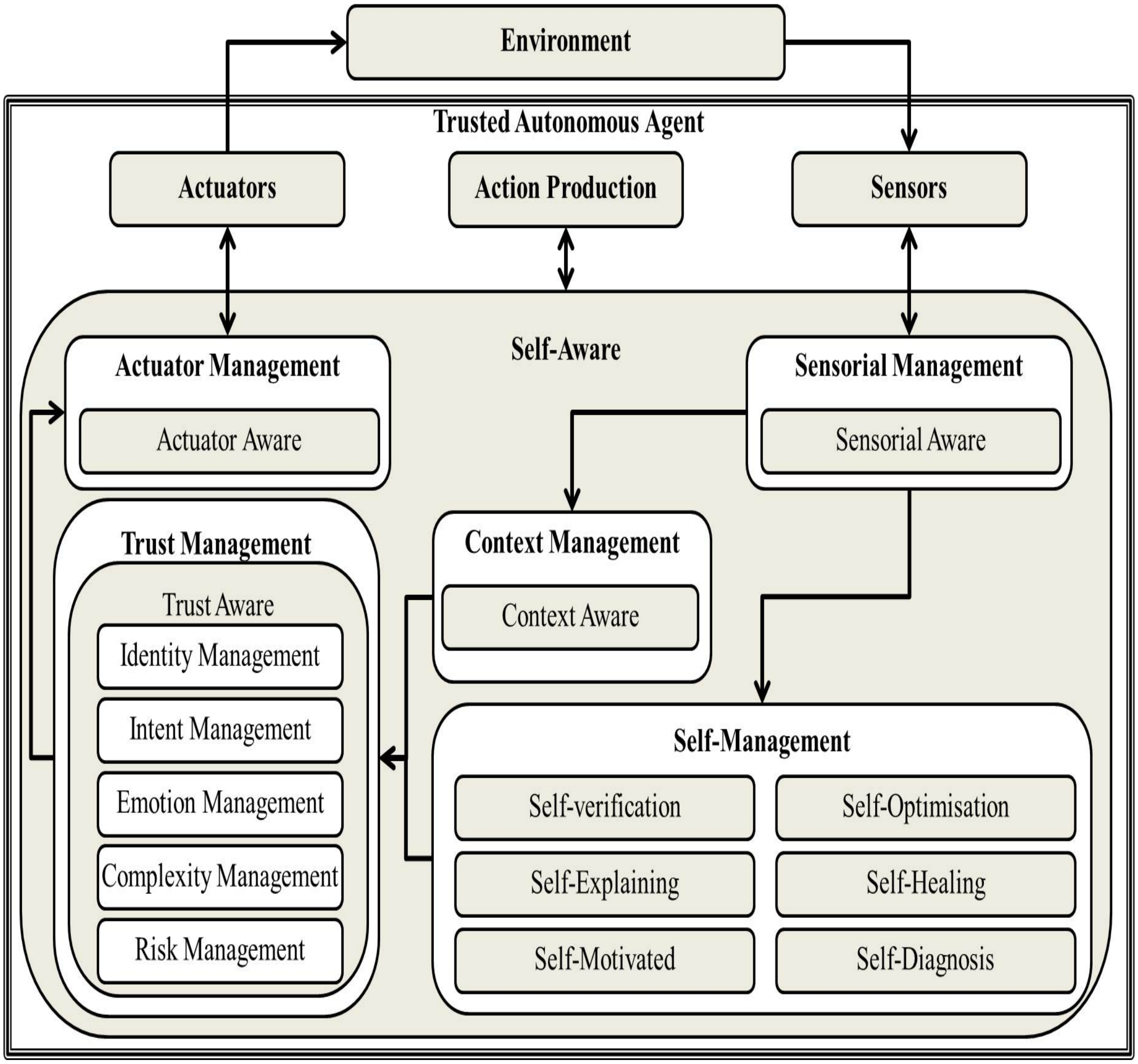

Fig. 6: Trust-Aware Autonomous Systems.

For a TAS to be trusted from a system-level perspective, it needs to be self-aware of system-level objectives and work collaboratively with other trusted-autonomous systems towards achieving these objectives.

The dynamics associated with the interaction require continuous sensing, processing, acting on, and verifying the environment it is operating within and other autonomous systems, including humans, in this environment. Verification [205], in particular, raises more challenges and requires a significant amount of data in real-time operations.

Figure 6 attempts to capture the key building blocks to form a TAS. The visual decomposition of these building blocks is primarily for ease of understanding and is not meant to imply that these building blocks can be designed or developed independently. It is a conceptual functional decomposition of the system, but it is not a proposal for a modular decomposition because of complexities that are outside the scope of this paper.

It might seem counterintuitive to separate action production from all modules that are focusing on self-management and self-awareness. This is intentional conceptually because action production mechanisms present a toolbox that any of the individual building blocks can use and adopt when needed and as the situation dictates. One can see action production as a service that gets called upon by all other modules.

Secondly, we adopt a convention that self-management will require self-awareness. We define self-awareness here as the system's ability to map out and represent all its components virtually (*ie.* software wise), continuously measure their states, and is able to identify if a component has been spoofed. This is a stricter definition of self-awareness than some of the general discussions of the concept in the literature. However, we take it that a definition need to be self-encompassing, while systems can exhibits different degrees of the concept being defined.

The following sub-section will highlight the big data challenges for trusted autonomy.

C. Big Data Challenges for TAS Revealed

Trusted autonomy creates a series of big data challenges. We will highlight below the six most prominent challenges that we have identified.

1) **Moving Sensor Network Challenge:** An autonomous system is a moving sensor network that collects data continuously, pro-actively, and in many instances intentionally to improve its performance and achieve its mission's objectives.

   We will put aside the resource-constrained environment of an autonomous platform – be it a hardware or a software agent, since this will be the challenge we will discuss next. For now, we will focus on the relationship between the agent's sensors and the environment.

   Many sensors will need to operate continuously. They can be operating even when they are not used by the system. This is particularly the case when a group of sensors are operating on the same switch; thus, switching one of them off would simply switch every sensor connected to the switch off. This raises an attractive offer for the wider context that the autonomous agent is operating within to leverage these data.

   For example, while a Google car is moving, it may have a temperature sensor to adjust the air conditioning system inside the car. If the air conditioning system is switched off, this temperature sensor can still be useful to transmit to the intelligent transportation network localised temperature information that can be integrated across the network to offer a real-time environmental picture of the transportation network.

   The above example imposes a big data challenge, when to allow for this data to be transmitted and when to prohibit transmission? On the one hand, transmitting the data benefits the wider system. On the other hand, it overloads the car's battery as a result of the extra energy required to transmit these data.

   Moreover, it opens trust challenges. The local temperature of the car may get seen by some as a not-threatening privacy data and get transmitted to third parties. A situation that can create security and privacy hazards, whereby the localised temperature can be used as a heat-signature for tracking the car by an entity that does have a privilege to do so and may not be allowed to recieve GPS information from the car.

   A second factor to consider in this challenge is related to the words 'pro-actively and intentionally'. The autonomous agent needs to decide which data is useful for its own context. In a dynamic environment where the situation is changing continuously, to decide which data is more useful to get, the autonomous entity needs to continuously be listening to all its sensors, processing the information and making a decision on the 'value' of the data so that it is able to pro-actively decide which sensor needs to be targeted for more data and to intentionally decide which decision is a higher priority at this point of time and the data needed for this decision.

   The problem above is very complex. To allow the system to continuously monitor all of its sensors, the system may need to reduce the sample rate for those sensors that may not be important at one point of time, then dynamically adjust the sampling rate as they become important. This is a very challenging problem.

   To automatically adapt the data acquisition strategy in a sensor network raises all sorts of big data problems, from the fact that the data arrives with different level of noise, the environment is dynamic and concepts change over time, to issues of sensor reliability and the trust-level associated with each sensor.

2) **Load-Balancing and Value Challenge:** Almost all TASs would have physical constraints in terms of memory and processing capacity. This resource-constrained environment compounds the complexity of too much available data, and the too little knowledge that we can draw out of it.

   A TAS, similar to a human, encounters data as it is operating that may not be useful for itself, but for others in the environment. Imagine two robots, one is collecting tomatoes and the other is collecting weeds. The robot collecting the tomatoes may sense the existence of weeds. Whether or not it should send this information to the other robot or not would depend on many factors. However, even the decision itself raises the question of how to balance the load on its memory, processing capacity, and battery life in these situations?

   The above question is a classic question in autonomous systems, but big data makes the question even more demanding. The data the TAS collects, even unintentionally, may hide a great deal of values to the system. The amount of processing and resources to be allocated for this task necessitate a level of processing that can threaten the survivability of the system in a harsh operating environment. Nevertheless, ignoring this information can threaten even more the survivability of this system.

3) **Trust-Awareness Cognitive Challenge:** For an autonomous system to be trusted, it can't focus purely on its primary mission as it needs to make every interaction it is involved within as part of its mission. This increases the data load on the system and creates higher than ever demands on the system to process these data in real-time.

   The primary challenge here is related to velocity, veracity and value. An agent that is acting in a real-time environment needs to mine the data it is receiving very quickly so that its actions take into account any patterns that have arisen in the moments before actions are made. A real-time situation is dynamic by nature. The changing situation needs to integrate its current states with new information. This integration process can be very expensive. On the one level it is a classic information fusion and belief update problem. On another level, it can trigger the activation of almost every component of a self-governing system. For example, the system may realise it needs to self-optimize in response to a self-diagnosed failure in one of its sensors, while simultaneously the self-healing module is attempting to recover the data through self-organising the rest of the sensor network. The synchronization of process and action in real-time is one of the grand challenges of TAS.

4) **Trusted Swarm Challenge:** A TAS does not work in isolation of other autonomous systems in the environment. Trust offer a path for autonomous systems to manage the complexities discussed above in the same way trust reduces complexity in a social system. It allows for delegation as a mechanism to manage complexity and distribute load among the system's components.

   In the previous challenge, we discussed an example which involved two types of complexity that the agent is faced with. One is related to the amount of processing an agent needs to do to synchronise its action production with a fast interaction. The second is related to loss of information as a result of sensor failure or a conscious decision by the agent. In any of these situations, if the autonomous system trusts another autonomous system, the former can use the latter to overcome some of these challenges, assuming that the latter can do the job.

   The above example focuses on a two-entity interaction, but the true benefits from trust becomes clearer when it happens within a swarm, whereby a whole lot of delegations can occur in the system among the trusted autonomous agents so that overall mission success is maximized. An example of this would be apparent in the mining industry where a swarm of machines need to cooperate to maximize the probability of discovering a suitable place to mine.

   The degree of coordination and delegation that occur in this example dramatically increase the data flow among the agents, whether these data are communicated directly through direct communications channels or indirectly through behavioural changes to signal a need for help. Again, this problem raises big data challenges because of value, velocity and veracity. There is a significant value add for the agents to cooperate in situations such as those described above to maximize the overall mission success of the swarm. These interactions are not free of complexity as the situation can be changing fast introducing a great deal of velocity challenges, and the fast sampling rate can be an extreme source of veracity which can cause instability in any inferring mechanism used in the system.

5) **Trusted Cyborg Challenge:** An artificial TAS – such as a software agent or a robot – is expected to work in harmony with humans. With appropriate use of advanced AI techniques, the artificial and human entities need to work as a single entity, an interdependency that is known as symbiosis.

   The challenges arises on how to make the silicon and biological agents work together as one; that is, establishing a truly symbiotic relationship to allow the machine and the human to meld as one entity. The coupling of the two creates a new type of agents, a Cyborg (human-machine coupled computational decision making engine).

   The machine needs to understand the human. However, the complexity arising from the diversity of human attitudes and personalities is huge. One possible mechanism to manage this complexity is to directly obtain objective data on the human to manage the ambiguity that comes with human communication. The data needs to be collected through mechanisms that do not interfere with the performance of the human on the job. The use of body sensors and passive external sensors such as cameras and microphones can offer an unprecedented amount of data to allow this integration to occur.

   The Cyborg's brain needs to have efficient interfaces between the biological and silicon brains so that both human and machine work seamlessly together as a single entity. We are not suggesting here to connect human brains to silicon or any type of science fiction. The suggestion is to integrate the decision making between the two in the same way two close friends work together to solve a problem.

   The Cyborg's brain needs on-board data processing. The real-time and volume of the data necessitate a very efficient use of the hardware. Technologies similar to those discussed in Section III such as neuromorphic computing, FPGA and other implementations of neural networks on chips can be very attractive. Equally important here is the need for real-time signal processing techniques capable of sensing the context to improve signal cleaning, and adapt sampling rate when and as needed, with appropriate dynamic calibration models that can fuse different sources of data from the human and dynamically adapt the parameters and structure of the model in response to changes in the environment.

6) **Trusted Cyborg Swarm Challenge:** A group of Cyborgs make a Cyborg swarm. The complexity of a network of Cyborg swarms, an extension of a problem we called in our previous research as Cognitive Cyber Symbiosis (CoCyS) [12], [55], needs to be managed to allow this decision network of Cyborgs to operate efficiently. CoCyS is a decision network of humans and machines. A Cyborg swarm is a network where each node is a Cyborg's brain.

   A Cyborg swarm is the next generation swarm intelligence system, where each human can look after a swarm of artificial TASs, and the Cyborg population works together as a swarm to solve far reaching complex.

   Take for example a situation like a post-disaster recovery, where a tsunami has destroyed a village. Each site may have a Cyborg consisting of a human and a swarm of artificial TASs. The Cyborg network needs to coordinate activities, share information, and work as one team to assist the village in recovery. Such a network should recover from the disaster a few orders of magnitude faster than what we do today, as a result of the close integration of human and machine.

   The challenges of forming such a network define scientific frontiers of this century. On the one hand, we have computational challenges to design appropriate AI systems that can handle this level of distributed control and complex and dynamic context. On the other hand, the amount of data in a Cyborg swarm is huge. These data are not just useful for the swarm to make decisions for recovery. They are also useful for future disaster situations, where it is important that these data get collected from the swarm as well.

Putting the six challenges above together, theoretical and practical aspects of the big data challenges for trusted autonomy draw a roadmap for the research needed to overcome these problems in practice.

References

[1] M. R. Delgado, R. H. Frank, and E. A. Phelps, "Perceptions of moral character modulate the neural systems of reward during the trust game," *Nature neuroscience*, vol. 8, no. 11, pp. 1611–1618, 2005.

[2] M. Deutsch, *The resolution of conflict: Constructive and destructive processes*. Yale University Press, 1977.

[3] ——, "Cooperation and trust: Some theoretical notes." in *Proceedings of Nebraska Symposium on Motivation*, M. R. Jones, Ed. Univer. Nebraska Press, 1962.

[4] N. Luhmann, H. Davis, J. Raffan, and K. Rooney, *Trust; and, Power: two works by Niklas Luhmann*. Wiley Chichester, 1979.

[5] C. Wells, "Trust, gender, and race in the workplace," *Journal of Social Behavior and Personality*, vol. 16, no. 1, pp. 115–126, 2001.

[6] D. S. Reina and M. L. Reina, "Trust and betrayal in the workplace building effective relationships in your organization," *Advances in Developing Human Resources*, vol. 2, no. 1, pp. 121–121, 2000.

[7] E. M. Whitener, S. E. Brodt, M. A. Korsgaard, and J. M. Werner, "Managers as initiators of trust: An exchange relationship framework for understanding managerial trustworthy behavior," *Academy of management review*, vol. 23, no. 3, pp. 513–530, 1998.

[8] M. D. Spector and G. E. Jones, "Trust in the workplace: Factors affecting trust formation between team members," *The Journal of social psychology*, vol. 144, no. 3, pp. 311–321, 2004.

[9] S. Marsh, "Formalising trust as a computational concept," PhD Thesis, 1994.

[10] I. Pinyol and J. Sabater-Mir, "Computational trust and reputation models for open multi-agent systems: a review," *Artificial Intelligence Review*, vol. 40, pp. 1–25, 2013.

[11] E. Petraki and H. Abbass, "On trust and influence: A computational red teaming game theoretic perspective," in *Computational Intelligence for Security and Defense Applications (CISDA), 2014 Seventh IEEE Symposium on*. IEEE, 2014, pp. 1–7.

[12] H. A. Abbass, E. Petraki, K. Merrick, J. Harvey, and M. Barlow, "Trusted autonomy and cognitive cyber symbiosis: Open challenges," *Cognitive Computation*, pp. 1–24, 2015.

[13] H. Abbass, G. Greenwood, and E. Petraki, "The n-player trust game and its replicator dynamics," *Evolutionary Computation, IEEE Transactions on*, To Appear.

[14] F. F. Alruwaili and T. A. Gulliver, "Secsdlc: A practical life cycle approach for cloud-based information security," *IJRCCT*, vol. 4, no. 2, pp. 095–107, 2015.

[15] J. Hinchman, M. Clark, J. Hoffman, B. Hulbert, A. Subcontractor, and C. Snyder, "Towards safety assurance of trusted autonomy in air force flight critical systems," in *Computer Security Applications Conference, Layered Assurance Workshop*, 2012.

[16] J. Sabater and C. Sierra, "Review on computational trust and reputation models," *Artificial Intelligence Review*, vol. 24, no. 1, pp. 33–60, 2005.

[17] A. Josang, R. Ismail, and C. Boyd, "A survey on trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[18] H. Yu, Z. Shen, C. Leung, C. Miao, and V. Lesser, "A survey of multi-agent trust management systems," *IEEE Access*, vol. 1, pp. 35–50, 2013.

[19] S. Marsh, "Optimism and pessimism in trust." The Ibero-American Conference on Artificial Intelligence, 1994, Conference Paper.

[20] N. Griffiths and M. Luck, "Coalition formation through motivation and trust." AAMAS 2003, 2003, Conference Paper, pp. 17–24.

[21] W. T. Luke Teacy, J. Patel, N. Jennings, and M. Luck, "Travos: trust and reputation in the context of inaccurate information sources," *Autonomous Agents and Multi Agent Systems*, vol. 12, pp. 183–198, 2006.

[22] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities." The Hawaii International Conference on System Sciences, 2000, Conference Paper.

[23] V. Cahill, E. Gray, M. Seigneur, C. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, G. di Marzo Serugendo, C. Bryce, M. Carbone, K. Krukow, and M. Nielsen, "Using trust for secure collaboration in uncertain environments," *Pervasive Computing*, vol. 2, no. 3, pp. 52–61, 2003.

[24] A. Josang, "Trust-based decision making for electronic transactions." The Fourth Nordic Workshop on Secure Computer Systems, 1999, Conference Paper.

[25] N. Griffiths, "A fuzzy approach to reasoning with trust, distrust and insufficient trust." CIA 2006, 2006, Conference Paper, pp. 360–374.

[26] A. Aref and T. Tran, "Using fuzzy logic and q-learning for trust modelling in multi-agent systems." The 2014 Federated Conference on Computer Science and Information Systems, 2014, Conference Paper, pp. 59–66.

[27] R. Levien, *Attack-resistant trust metrics*, 2009, pp. 121–132.

[28] C. N. Ziegler and G. Lausen, "Spreading activation models for trust propagation." The 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004, Conference Paper, pp. 83–97.

[29] G. Mahoney, W. Myrvold, and G. Shoja, "Generic reliability trust model," in *The Third Annual Conference on Privacy, Security and Trust*, 2005.

[30] A. Conn, N. Gould, and P. Toint, *Trust-region methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics and Mathematical Programming Society, 2000.

[31] Y. Yuan, "A review of trust region algorithms for optimisation." Edinburgh: The Fourth International Congress on Industrial and Applied Mathematics, 1999.

[32] Z. Giorgos, "Collaborative reputation mechanisms for online communities," Masters Thesis, 1999.

[33] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 843–857, 2004.

[34] J. Hu, Q. Wu, and B. Zhou, "Fctrust: A robust and efficient feedback credibility-based distributed p2p trust model," in *The Ninth IEEE International Conference for Young Computer Scientiests*, 2008.

[35] Y. Zhang, S. Chen, and G. Yang, "Sftrust: a double trust metric based trust model in unstructured p2p systems," in *The IEEE International Symposium on Parallel and Distributed Processing*, 2009.

[36] C. M. Jonker and J. Treur, "Formal analysis of models for the dynamics of trust based on experiences." MAAMAW 99, 1999, Conference Paper, pp. 120–130.

[37] X. Fan, S. Oh, M. McNeese, J. Yen, H. Cuevas, L. Strater, and M. Endsley, "The influence of agent reliability on trust in human-agent collaboration," in *The fifteenth european conference on cognitive ergonomics*, 2008.

[38] M. Wooldridge, *Reasoning about rational agents*. Cambridge, MA: The MIT Press, 2000.

[39] B. Jarvis and L. Jain, "Trust in lora: towards a formal definition of trust in bdi agents." KES 2006, 2006, Conference Paper, pp. 458–463.

[40] T. Taibi, "Incorporating trust into the bdi architecture," *International Journal of Artificial Intelligence and Soft Computing*, vol. 2, no. 3, pp. 223–230, 2010.

[41] A. Koster, M. Scholrlemmer, and J. Sabater-Mir, “Opening the black box of trust: reasoning about trust models in a bdi agent,” *Journal of Logic and Computation*, vol. 23, no. 1, pp. 25–58, 2013.

[42] G. Xu, C. Xu, X. Tian, L. Zhang, X. Li, and W. Li, “Pso-tps: An optimal trust path selection algorithm based on particle swarm optimisation in small world network,” in *The Second International Conference on Cloud and Green Computing*, 2012.

[43] W. Huang, Z. Deng, R. Li, and X. Tang, “Trust-based particle swarm optimisation for grid task scheduling,” *Applied Mechanics and Materials*, vol. 239-240, pp. 1331–1335, 2013.

[44] K. Merrick and M. L. Maher, *Motivated reinforcement learning: curious characters for multiuser games*. Berlin: Springer, 2009.

[45] C. Burnett, T. Norman, and K. Sycara. The 22nd International Joint Conference on Artificial Intelligence, 2011, Conference Paper, pp. 115–120.

[46] C. M. Jonker, S. Meijer, D. Tykhonov, and T. Verwaart, “Modelling and simulation of selling and deceit for the trust and tracing game.” The Trust in Agent Societies Workshop, 2005, Conference Paper, pp. 78–90.

[47] Z. Sun, G. Finnie, and J. Barker, “Trust and deception in multi-agent trading systems: a logical viewpoint.” The 11th Americas Conference on Information Systems, 2005, Conference Paper, pp. pp 1020–1026.

[48] G. J. Hofstede, C. M. Jonker, S. Meijer, and T. Verwaart, “Modelling trade and trust across cultures.” The Fourth International Conference on Trust Management and iTrust, 2006, Conference Paper, pp. 120–134.

[49] G. J. Hofstede, C. M. Jonker, and T. Verwaart, “A multi-agent model of deceit and trust in intercultural trust.” The International Conference on Computational Collective Intelligence, 2009, Conference Paper.

[50] A. Das, M. M. Islam, and G. Sorwar, “Dynamic trust model for reliable transactions in multi-agent systems,” in *The Thirteenth International Conference on Advanced Communication Technology*, 2011.

[51] K. Fullam, T. Klos, G. Muller, J. Sabater, A. Scholosser, Z. Topol, S. Barber, J. Rosenschein, L. Vercouter, and M. Voss, “A specification of the agent reputation and trust (art) testbed: experimentation and competitition for trust in agent societies.” Autonomous Agents and Multi Agent Systems, 2005, Conference Paper, pp. 512–518.

[52] G. Leu and H. A. Abbass, “A multi-disciplinary review of knowledge acquisition methods: From human to autonomous eliciting agents,” *Knowledge-Based Systems*, To Appear.

[53] P. Russom *et al.*, “Big data analytics,” *TDWI Best Practices Report, Fourth Quarter*, 2011.

[54] Y. Jin and B. Hammer, “Computational intelligence in big data [guest editorial],” *Computational Intelligence Magazine, IEEE*, vol. 9, no. 3, pp. 12–13, 2014.

[55] H. Abbass, *Computational Red Teaming: Risk Analytics of Big-Data-to-Decisions Intelligent Systems*. Springer Verlag, 2015.

[56] C. Moron, C. Cabrera, A. Moron, A. Garcia, and M. Gonzalez, “Magnetic sensors based on amorphous ferromagnetic materials: A review,” *Sensors*, vol. 15, no. 11, pp. 28 340–28 366, 2015.

[57] C. Bisdikian, “On sensor sampling and quality of information: A starting point,” in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops ’07. Fifth Annual IEEE International Conference on*, March 2007, pp. 279–284.

[58] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, “A review of wearable sensors and systems with application in rehabilitation,” *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 1, pp. 1–17, 2012. [Online]. Available: http://dx.doi.org/10.1186/1743-0003-9-21

[59] S. C. Mukhopadhyay, “Wearable sensors for human activity monitoring: A review,” *Sensors Journal, IEEE*, vol. 15, no. 3, pp. 1321–1330, March 2015.

[60] Athos. (2016, 02) The world’s first smart fitness apparel that measures muscle activity and heart rate all in real time. accessed 24.02.2016. [Online]. Available: https://www.liveathos.com/

[61] H. Banaee, M. U. Ahmed, and A. Loutfi, “Data mining for wearable sensors in health monitoring systems: A review of recent trends and challenges,” *Sensors*, vol. 13, no. 12, pp. 17 472–17 500, 2013. [Online]. Available: http://www.mdpi.com/1424-8220/13/12/17472

[62] M. Mirzaei and M. Sawan, “Microelectronics-based biosensors dedicated to the detection of neurotransmitters: A review,” *Sensors*, vol. 14, no. 10, pp. 17 981–18 008, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/10/17981

[63] A. J. Bandodkar and J. Wang, “Non-invasive wearable electrochemical sensors: a review,” *Trends in Biotechnology*, vol. 32, no. 7, pp. 363 – 371, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167779914000699

[64] M. Stoppa and A. Chiolerio, “Wearable electronics and smart textiles: A critical review,” *Sensors*, vol. 14, no. 7, p. 11957, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/7/11957

[65] S. Sprager and M. B. Juric, “Inertial sensor-based gait recognition: A review,” *Sensors*, vol. 15, no. 9, p. 22089, 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/9/22089

[66] J. Jeon, H.-B.-R. Lee, and Z. Bao, “Flexible wireless temperature sensors based on ni microparticle-filled binary polymer composites,” *Advanced Materials*, vol. 25, no. 6, pp. 850–855, 2013.

[67] Beurer. (2016) Ft90 - contactless temperature sensor. accessed 26.02.2016. [Online]. Available: https://www.beurer.com/web/en/products/body_temperature/body_temperature/FT-90

[68] F. Kovacs, M. Torok, and I. Habermajer, “A rule-based phonocardiographic method for long-term fetal heart rate monitoring,” *Biomedical Engineering, IEEE Transactions on*, vol. 47, no. 1, pp. 124–130, Jan 2000.

[69] Q. Zhu, J. Spinelli, and G. Carlson, “Accelerometer-based heart sound detection for autocapture,” Nov. 18 2003, uS Patent 6,650,940. [Online]. Available: http://www.google.com/patents/US6650940

[70] A. Alzahrani, S. Hu, V. Azorin-Peris, L. Barrett, D. Esliger, M. Hayes, S. Akbare, J. Achart, and S. Kuoch, “A multi-channel opto-electronic sensor to accurately monitor heart rate against motion artefact during exercise,” *Sensors*, vol. 15, no. 10, p. 25681, 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/10/25681

[71] S. M. A. Salehizadeh, D. Dao, J. Bolkhovsky, C. Cho, Y. Mendelson, and K. H. Chon, “A novel time-varying spectral filtering algorithm for reconstruction of motion artifact corrupted heart rate signals during intense physical activities using a wearable photoplethysmogram sensor,” *Sensors*, vol. 16, no. 1, p. 10, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/1/10

[72] J.-H. Park, D.-G. Jang, J. W. Park, and S.-K. Youm, “Wearable sensing of in-ear pressure for heart rate monitoring with a piezoelectric sensor,” *Sensors*, vol. 15, no. 9, p. 23402, 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/9/23402

[73] S. Garmin. (2016) accessed 24.02.2016. [Online]. Available: http://www.dcrainmaker.com/2015/11/garmin-hrm-swimtri-review.html

[74] RAF. (2016) Optical wrist based heart rate versus chest straps. accessed 24.02.2016. [Online]. Available: http://www.shutuplegs.org/optical-hr-versus-chest-strap/

[75] G. EGI. (2016, 02) Clinical geodesic eeg system 400. accessed 22.02.2016. [Online]. Available: http://www.egi.com/clinical-division/clinical-division-clinical-products/ges-400-series

[76] H. A. Abbass, J. Tang, R. Amin, M. Ellejmi, and S. Kirby, “Augmented cognition using real-time eeg-based adaptive strategies for air traffic control,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1, pp. 230–234, 2014.

[77] H. A. Abbass, J. Tang, M. Ellejmi, and S. Kirby, “Visual and auditory reaction time for air traffic controllers using quantitative electroencephalograph (qeeg) data,” *Brain Informatics*, vol. 1, no. 1, pp. 39–45, 2014.

[78] I. Emotiv. (2016, 02) Emotiv insight - brain activity tracker. accessed 22.02.2016. [Online]. Available: https://emotiv.com/insight.php

[79] E. Emotiv. (2016, 02) A high resolution, multi-channel eeg system. accessed 22.02.2016. [Online]. Available: https://emotiv.com/epoc.php

[80] T. Nihonkohden. (2016, 02) Trackit - mobile long-term eeg. accessed 22.02.2016. [Online]. Available: http://www.nihonkohden.de/products/neurology/eeg/mobile-long-term-eeg/trackit.html?L=1

[81] S. Compumedics. (2016, 02) Synamps 2/rt 256-channel eeg/ep/erp. accessed 22.02.2016. [Online]. Available: http://compumedicsneuroscan.com/synamps-rt-256-channel-eegeperp/

[82] eegoMylab antNeuro. (2016, 02) eego mylab - the new frontier in multimodal brain research. accessed 22.02.2016. [Online]. Available: http://www.ant-neuro.com/products/eego_mylab

[83] G. Compumedics. (2016, 02) Grael 32-channel eeg/erp amplifier. accessed 22.02.2016. [Online]. Available: http://compumedicsneuroscan.com/grael-eegerp-amplifier/

[84] N. MindMedia. (2016) Nexus-32 multimodal qeeg system. accessed 04.03.2016. [Online]. Available: http://www.mindmedia.info/CMS2014/products/systems/nexus-32

[85] Mitsar. (2016, 02) Mitsar eeg 201/202 compare. accessed 22.02.2016. [Online]. Available: http://www.mitsar-medical.com/eeg-machine/eeg-amplifier-compare/

[86] N. Nohonkohden. (2016, 02) Neurofax eeg-1200. accessed 22.02.2016. [Online]. Available: http://www.nihonkohden.de/products/neurology/eeg/research/eeg-1200.html?L=1

[87] B. Neurovirtual. (2016, 02) Bwiii eeg plus icu brain monitor. accessed 22.02.2016. [Online]. Available: http://neurovirtual.com/equipment/bwiii-eeg-plus-ltmicu/

[88] T. Neuroconn. (2016, 02) Measuring and modulating brain activity. accessed 22.02.2016. [Online]. Available: http://www.neuroconn.de/thera_prax_q-eeg_en/

[89] G. EGI. (2016, 02) Clinical geodesic eeg mobile (gem) 100. accessed 22.02.2016. [Online]. Available: http://www.egi.com/clinical-division/clinical-division-clinical-products/clinical-division-geodesic-eeg-mobile-100

[90] encephalan Medicom. (2016, 02) High-quality eeg recording and freedom of movement. accessed 22.02.2016. [Online]. Available: http://www.medicom-mtd.com/eng/Products/eegr.htm

[91] Neurowerk. (2016, 02) Neurowerk eeg systems. accessed 22.02.2016. [Online]. Available: http://www.neurowerk.de/fileadmin/user_upload/downloads_english/NEUROWERK_EEG_EN.pdf

[92] nuamps compumedics. (2016, 02) Nuamps 40-channel eeg/erp amplifier. accessed 22.02.2016. [Online]. Available: http://compumedicsneuroscan.com/nuamps-eegerp-amplifier/

[93] T. Deymed. (2016, 02) Advanced eeg system built for demanding clinical and research use. accessed 22.02.2016. [Online]. Available: http://www.deymed.com/products-a-services/truscan-eeg/product-info

[94] eegoSports antNeuro. (2016, 02) eego sports - the total mobility solution. accessed 22.02.2016. [Online]. Available: http://www.ant-neuro.com/products/eego_sports

[95] B. Neurovirtual. (2016, 02) Bwmini ambulatory eeg. accessed 22.02.2016. [Online]. Available: http://neurovirtual.com/equipment/bwmini-ambulatory-eeg/

[96] M. NeuroSky. (2016, 02) Neurosky mindwave mobile headset kit. accessed 22.02.2016. [Online]. Available: http://neurosky.com/biosensors/eeg-sensor/

[97] M. Louwerse and S. Hutchinson, "Neurological evidence linguistic processes precede perceptual simulation in conceptual processing," *Frontiers in Psychology*, vol. 3, no. 385, 2012. [Online]. Available: http://www.frontiersin.org/cognitive_science/10.3389/fpsyg.2012.00385/abstract

[98] T. Fraga, M. Pichiliani, and D. Louro, *Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for eInclusion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ch. Experimental Art with Brain Controlled Interface, pp. 642–651.

[99] B. Macrotellect. (2016, 02) accessed 22.02.2016. [Online]. Available: http://o.macrotellect.com/en/brainlink/principle.aspx

[100] S. Charara. (2015, 04) Muse review: The brain sensing headband that knows you're stressed. accessed on 22.02.2016. www.wearable.com. [Online]. Available: http://www.wareable.com/wearable-tech/muse-brain-sensing-headband-review-938

[101] M. Interaxon. (2016, 02) Muse - the brain sensing headband. accessed on 22.02.2016. [Online]. Available: http://www.choosemuse.com/what-does-muse-measure/

[102] Somnomedics. (2016, 02) Technical data - somnoscreen eeg 32. accessed 22.02.2016. [Online]. Available: http://somnomedics.eu/products/neurologic-applications-up-to-58-channels/technical-data-somnoscreentm-plus-eeg-32/

[103] Neurosoft. (2016, 02) Neuron spectrum am - ambulatory wireless eeg/psg recorder. accessed 22.02.2016. [Online]. Available: http://neurosoft.com/en/catalog/view/id/2107/sid/1

[104] B. Deymed. (2016, 02) Clinical grade neurofeedback system dedesign for clinicians wishing to use the highigh standard for neurotherapy. accessed 22.02.2016. [Online]. Available: http://www.deymed.com/products-a-services/bfb-pro/specifications

[105] A. Cardio02line. (2016, 02) Portable electrocardiograph with thermal printing system. accessed 22.02.2016. [Online]. Available: http://eng.cardioline.com/prodotti-cardioline/ar600adv/

[106] T. Cardioline. (2016, 02) Cardioline touchecg tablet hd+. accessed 22.02.2016. [Online]. Available: http://eng.cardioline.com/prodotti-cardioline/touchecg-tablet-hd/

[107] U. TSE. (2016, 02) Electrocardiograph ucard - 200. accessed 22.02.2016. [Online]. Available: http://www.tse.cz/images/stories/download/9.pdf

[108] S. Nasan. (2016, 02) Pc based ecg machine with simultaneous acquisition. accessed 22.02.2016. [Online]. Available: http://www.nasanmedical.com/simul-g.html

[109] E. Longfian. (2016, 02) Monochrome 12 channel ecg. accessed 22.02.2016. [Online]. Available: http://www.longfian.com/en/product/product37.html

[110] E.-G. Nasan. (2016, 02) Dsp based single channel ecg machine. accessed 22.02.2016. [Online]. Available: http://www.nasanmedical.com/esy-g1.html

[111] A. Nasan. (2016, 02) 3 channel ecg machine with measurement interpretation. accessed 22.02.2016. [Online]. Available: http://www.nasanmedical.com/a-saan-1003.html

[112] R. Nasiff. (2016, 02) Nasiff cardiocard pc based resting ecg system. accessed 22.02.2016. [Online]. Available: https://nasiff.com/resting.html

[113] C. SpaceLab. (2016, 02) Cardiodirect 12 usb. accessed 22.02.2016. [Online]. Available: http://www.spacelabshealthcare.com/diagnostic-cardiology/resting-ecg/cardiodirect-12-usb#.VsrctmoViOV

[114] C. Customed. (2016, 92) Custocardio 100/110 usb. accessed 22.02.2016. [Online]. Available: https://www.customed.de/includes/customed%20documents/TFL/TFL-0007-custo-cardio-100-110-USB-24-03-2010-DK-0963-DE-004-LEN.pdf

[115] K. Kalamed. (2016, 02) High-end 12-channel electrocardiograph for resting and stress ecg. accessed 22.02.2016. [Online]. Available: http://www.kalamed.de/en-us/produkte/ekg/kes-1211.aspx

[116] ——. (2016, 02) High quality 12-ch ecg-recorder with usb or wifi connection to pc or android app. accessed 22.02.2016. [Online]. Available: http://www.kalamed.de/en-us/produkte/pc-ekg/kec-1000.aspx

[117] Labtech. (2016, 02) Ec-12rm - resting smartphone-based wireless electrocardiograph. accessed 22.02.2016. [Online]. Available: http://www.labtech.hu/products/netecg/ec-12rm

[118] C. Customed02. (2016) Custo cardio 200 usb/bt. accessed 22.02.2016. [Online]. Available: https://www.customed.de/includes/customed%20documents/TFL/TFL-0011-custo-cardio-200-USB-BT-24-03-2010-DK-0967-DE-004-LEN.pdf

[119] E.-V. Eccosur. (2016, 02) Sistema de registro electrocardiografico de esfuerzo de 12 derivaciones basado en pc. accessed 23.02.2016. [Online]. Available: http://www.eccosur.com/product.php?inc=er

[120] T. Thor. (2016, 02) Thoriax - pc ecg. accessed 23.02.2016. [Online]. Available: http://thorlabor.com/en/products/cardiac-medical-devices/thoriax-pc-ecg/

[121] C. SpaceLab. (2016, 02) Cardiodirect 12s. accessed 22.02.2016. [Online]. Available: http://www.spacelabshealthcare.com/diagnostic-cardiology/stress-testing/cardiodirect-12s#.VsrfQWoViOV

[122] C. Cardioline. (2016, 02) Cubestress hd+ series. accessed 22.02.2016. [Online]. Available: http://eng.cardioline.com/prodotti-cardioline/system-cubestress-hd-mit/

[123] C. Cardionics. (2016, 02) The cartouch. accessed 23.02.2016. [Online]. Available: http://www.cardionics.eu/English/cartouchE.html

[124] E.-R. Labtech. (2016, 02) Ec-12rs 12 channel resting and stress test ecg system. accessed 23.02.2016. [Online]. Available: http://www.labtech.hu/products/resting-and-stress-ecg-systems/ec-12r-s.html

[125] C. Amedtec. (2016, 02) Cardiopart 12 blue/blue-p. accessed 23.02.2016. [Online]. Available: http://www.amedtec.de/downloads/CardioPart%2012%20Blue_en.pdf

[126] T. Thor. (2016, 02) Thoriax home ecg - portable ecg monitor. accessed 23.02.2016. [Online]. Available: http://thorlabor.com/en/products/cardiac-medical-devices/thoriax-home-ecg/

[127] W. Boucsein, *Electrodermal activity*. Springer Science & Business Media, 2012.

[128] R. W. Picard, S. Fedor, and Y. Ayzenberg, "Multiple arousal theory and daily-life electrodermal activity asymmetry," *Emotion Review*, vol. 8, no. 1, pp. 62–75, 2016. [Online]. Available: http://emr.sagepub.com/content/8/1/62.abstract

[129] Thoughttechnology. (2016, 02) Technical note series - skin conductance sensor (sa9309m). accessed 23.02.2016. [Online]. Available: http://www.thoughttechnology.com/sciencedivision/media/pdf/STN0008-00%20-%20%20SA9309M%20Skin%20Conductance%20Technical%20Notes.pdf

[130] W. B. Mendes, *Assessing autonomic nervous system activity*. Guilford Press, 2009, pp. 118–147.

[131] Empatica. (2016, 02) The embrace journey - from the idea through the crowdfunding to the final product. [Online]. Available: https://www.empatica.com/embrace-watch-about

[132] Neulog. (2016, 02) Neulog - gsr logger sensor nul-217. accessed 23.02.2016. [Online]. Available: https://neulog.com/gsr/

[133] Shimmersensing. (23.02.2016, 02) Shimmer3 gsr+ unit. [Online]. Available: http://www.shimmersensing.com/shop/shimmer3-wireless-gsr-sensor#specifications-tab

[134] Mindmedia. (2016, 02) Skin conductance sensor. accessed 23.02.2016. [Online]. Available: http://www.mindmedia.info/CMS2014/products/sensors/skin-conductance-sensor

[135] G. TMSI. (2016, 02) Gsr sensor. accessed 23.02.2016. [Online]. Available: http://www.tmsi.com/products/accessories/item/gsr-sensor

[136] Mindfield. (2016, 02) Mindfield esense skin response. accessed 23.02.2016. [Online]. Available: https://www.mindfield.de/en/biofeedback/products/esense/esense-skin-response

[137] Angelsensor. (2016, 02) Angelsenros. accessed 24.02.2016. [Online]. Available: http://angelsensor.com/

[138] A. Apple. (2016, 02) To wear it is to love it. accessed 24.02.2016. [Online]. Available: http://www.apple.com/watch/?cid=w-wa-us-kwg-watch-com

[139] B. Mybasis. (2016, 02) Basis peak - the ultimate fitness and sleep tracker. accessed 24.02.2016. [Online]. Available: http://www.mybasis.com/

[140] E. series Empatica. (2014) The e4 wristband is a wearable wireless device designed for continuous, real-time data acquisition in daily life. accessed 24.02.2016. [Online]. Available: https://www.empatica.com/e4-wristband

[141] Z. Fitbit. (2016, 02) Zip - wireless activity tracker. accessed 24.02.2016. [Online]. Available: https://www.fitbit.com/zip

[142] W. Fitbit. (2016, 02) The power of heart rate on your wrist. accessed 24.02.2016. [Online]. Available: https://www.fitbit.com/au/chargehr

[143] ——. (2016, 02) The ultimate fitness super watch. accessed 24.02.2016. [Online]. Available: https://www.fitbit.com/au/surge

[144] O. Fitbug. (2016, 02) Orb - fitness, sleep and activity tracker. accessed on 24.02.2016. [Online]. Available: https://www.fitbug.com/g/orb?lng=en_UK

[145] Garmin. (2016, 02) Meet the wearables that help you move a little more each day. accessed 24.02.2016. [Online]. Available: http://explore.garmin.com/en-US/vivo-fitness/

[146] A. Google. (2016, 02) Android wearable devices. accessed 24.02.2016. [Online]. Available: http://www.android.com/wear/

[147] iHealth. ihealth edge - take your fitness to the edge and beyond with this activity + sleep tracker. accessed 24.02.2016. [Online]. Available: https://ihealthlabs.com/fitness-devices/ihealth-edge/

[148] U. Jawbone. (2016) Up is a revolutionary system that guides you every step of the way to a better, healthier you. accessed 24.02.2016. [Online]. Available: https://jawbone.com/up

[149] L. LG. (2016, 02) Lg lifeband touch activity tracker - make the world your workout. accessed 24.02.2016. [Online]. Available: http://www.lg.com/au/fitness-and-wearables/lg-FB84-BM

[150] S. LG. (2016, 02) Lg g watch (w100) black titan - information when you need it most. accessed 24.02.2016. [Online]. Available: http://www.lg.com/au/fitness-and-wearables/lg-G-WATCH-W100-Black-Titan

[151] V. Medisana. (2016) Activity and sleep tracker in one. accessed 24.02.2016. [Online]. Available: http://www.medisana.com/en/Sport/Activity-Tracker/ViFit-touch-Activity-Tracker-black.html

[152] B. Microsoft. (2016) Live healthier and achieve more. accessed 24.02.2016. [Online]. Available: https://www.microsoft.com/microsoft-band/en-au

[153] Misfit. (2016, 02) Fitness and sleep monitor. accessed 24.02.2016. [Online]. Available: http://misfit.com/

[154] F. Nike. (2016, 02) Connect and go with nike+. accessed 24.02.2016. [Online]. Available: https://secure-nikeplus.nike.com/plus/products/fuelband/

[155] Pebble. (2016, 02) Pebble smartwasmart series. accessed 24.02.2016. [Online]. Available: https://www.pebble.com/compare

[156] N. Razerone. (2016, 02) Survive your world with the razer nabu. accessed 24.02.2016. [Online]. Available: http://www.razerzone.com/nabu

[157] W. Razerone. (2016, 02) Nabu watch - live smarter. accessed 24.02.2016. [Online]. Available: http://www.razerzone.com/nabu-watch

[158] L. Salutron. (2016) Achieve your health and fitness goals with lifetrak. accessed 24.02.2016. [Online]. Available: http://lifetrakusa.com/

[159] G. Samsung. (2016) accessed 24.02.2016. [Online]. Available: http://www.samsung.com/au/gear-s2/

[160] ——. (2016) accessed 24.02.2016. [Online]. Available: http://www.samsung.com/au/consumer/mobile-phone/wearables/gear/SM-R3500ZKAXSA

[161] Sensoriafitness. (2016, 02) A smarter way to run - improve speed, pace, cadence, foot landing and compare your shoes ? all with a sock! accessed 24.02.2016. [Online]. Available: http://www.sensoriafitness.com/

[162] S. Sony. (2016, 02) Live smarter with smartwear. accessed 24.02.2016. [Online]. Available: http://www.sonymobile.com/us/products/smartwear/

[163] F. Striiv. (2016, 02) Striiv - fusion series. accessed 24.02.2016. [Online]. Available: https://www.striiv.com/

[164] P. Withings. (2016) accessed 24.02.2016. [Online]. Available: http://www.withings.com/us/en/products/pulse?

[165] G. Withings. (2016) accessed 24.02.2016. [Online]. Available: http://www.withings.com/us/en/products/go

[166] W. Elmenreich, *Software Technologies for Embedded and Ubiquitous Systems: 5th IFIP WG 10.2 International Workshop, SEUS 2007, Santorini Island, Greece, May 2007. Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, ch. A Review on System Architectures for Sensor Fusion Applications, pp. 547–559. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-75664-4_57

[167] S. A. Khamseh, A. K. Sedigh, B. Moshiri, and A. Fatehi, "Control performance assessment based on sensor fusion techniques," *Control Engineering Practice*, vol. 49, pp. 14 – 28, 2016.

[168] C. Jiang, Y. C. Soh, and H. Li, "Sensor and {CFD} data fusion for airflow field estimation," *Applied Thermal Engineering*, vol. 92, pp. 149 – 161, 2016.

[169] D. Novak and R. Riener, "A survey of sensor fusion methods in wearable robotics," *Robotics and Autonomous Systems*, vol. 73, pp. 155 – 170, 2015, wearable Robotics.

[170] D. Tansky, A. Fischer, B. M. Colosimo, L. Pagani, and Y. B. Shabat, "Multi-sensor multi-resolution data fusion modeling," *Procedia {CIRP}*, vol. 21, pp. 151 – 158, 2014, 24th {CIRP} Design Conference.

[171] H. Durrant-White. (2002) Introduction to sensor data fusion. accessed 25.02.2016. [Online]. Available: http://www.acfr.usyd.edu.au/pdfs/training/IntroDataFusionSlides.pdf

[172] A. R. Mirza, "Data fusion architectures for sensor platforms," in *Aerospace Conference, 2008 IEEE*, March 2008, pp. 1–13.

[173] E. Azimirad, J. Haddadnia, and A. Izadipour, "A comprehensive review of the multi-sensor data fusion architectures," *Journal of Theoretical and Applied Information Technology*, vol. 71, no. 1, January 2015. [Online]. Available: http://www.jatit.org/volumes/Vol71No1/4Vol71No1.pdf

[174] B. V. Dasarathy, "Sensor fusion potential exploitation-innovative architectures and illustrative applications," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 24–38, Jan 1997.

[175] B. J. MacLennan, “Natural computation and non-turing models of computation,” *Theoretical Computer Science*, vol. 317, no. 1 - 3, pp. 115 – 145, 2004, super-Recursive Algorithms and Hypercomputation. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397503006352

[176] ——, *Encyclopedia of Complexity and Systems Science*. New York, NY: Springer New York, 2009, ch. Analog Computation, pp. 271–294. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-30440-3_19

[177] J. S. Small, “General-purpose electronic analog computing: 1945-1965,” *Annals of the History of Computing, IEEE*, vol. 15, no. 2, pp. 8–18, 1993.

[178] K. H. Lundberg, “The history of analog computing: introduction to the special section,” *Control Systems, IEEE*, vol. 25, no. 3, pp. 22–25, June 2005.

[179] A. E. Rogers and T. W. Connelly, *Analog computation in engineering design*. New York : Mcgraw-Hill, 1960.

[180] J. S. Small, *The analogue alternative: The electronic analogue computer in Britain and the USA, 1930-1975*. Routledge, 2001, vol. 15.

[181] P. Wambacq and W. Sansen, *Distortion analysis of analog integrated circuits*. Springer Science & Business Media, 1998, vol. 451.

[182] R. Daniel, J. R. Rubens, R. Sarpeshkar, and T. K. Lu, “Synthetic analog computation in living cells,” *Nature*, vol. 497, no. 7451, pp. 619–623, 2013. [Online]. Available: http://dx.doi.org/10.1038/nature12148

[183] G. E. R. Cowan, R. C. Melville, and Y. Tsividis, “A vlsi analog computer/math co-processor for a digital computer,” in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, Feb 2005, pp. 82–586 Vol. 1.

[184] B. Schell and Y. Tsividis, “A clockless adc/dsp/dac system with activity-dependent power dissipation and no aliasing,” in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, Feb 2008, pp. 550–635.

[185] N. Guo, Y. Huang, T. Mai, S. Patil, C. Cao, M. Seok, S. Sethumadhavan, and Y. Tsividis, “Continuous-time hybrid computation with programmable nonlinearities,” in *European Solid-State Circuits Conference (ESSCIRC), ESSCIRC 2015 - 41st*, Sept 2015, pp. 279–282.

[186] M. P. Singh and M. K. Jain, “Article: Evolution of processor architecture in mobile phones,” *International Journal of Computer Applications*, vol. 90, no. 4, pp. 34–39, March 2014.

[187] A. R. Brodtkorb, T. R. Hagen, and M. L. Saetra, “Graphics processing unit (gpu) programming strategies and trends in {GPU} computing,” *Journal of Parallel and Distributed Computing*, vol. 73, no. 1, pp. 4 – 13, 2013, metaheuristics on {GPUs}.

[188] H. Khatter and V. Aggarwal, “Efficient parallel processing by improved cpu-gpu interaction,” in *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*, Feb 2014, pp. 159–161.

[189] M. Microsoft. (2016, 02) Directx graphics and gaming. accessed on 19.02.2016. Micrisoft. [Online]. Available: https://msdn.microsoft.com/en-us/library/ee663274(v=vs.85).aspx

[190] nVidia. (2016, 02) Cuda parallel computing platform. accessed on 19.02.2016. [Online]. Available: http://www.nvidia.com/object/cuda_home_new.html

[191] A. AMD. (2016, 02) Amd compute cores. accessed on 19.02.2016. [Online]. Available: http://www.amd.com/en-us/innovations/software-technologies/processors-for-business/compute-cores

[192] O. Khronos. (2016, 02) The open standard for parallel programming of heterogeneous systems. accessed on 19.02.2016. [Online]. Available: https://www.khronos.org/opencl/

[193] R. Cumplido, E. de la Torre, C. Feregrino-Uribe, and M. Wirthlin, “Introduction to special issue on reconfigurable computing and {FPGAs},” *Microprocessors and Microsystems*, vol. 39, no. 7, pp. 541 – 542, 2015.

[194] H. Hassan and M. Anis, “Chapter 3 - power estimation in {FPGAs},” in *Low-Power Design of Nanometer {FPGAs}*, ser. Systems on Silicon, H. Hassan and M. Anis, Eds. Boston: Morgan Kaufmann, 2010, pp. 41 – 83.

[195] R. Lysecky and F. Vahid, “A study of the speedups and competitiveness of fpga soft processor cores using dynamic hardware/software partitioning.” *Design Automation and Test in Europe*, pp. 18–23, March 2005. [Online]. Available: http://www.cs.ucr.edu/~vahid/pubs/date05_warp_microblaze.pdf

[196] P. Ekas and B. Jentz, “Developing and integrating fpga coprocessors,” *Embedded computing design*, 2003. [Online]. Available: http://embedded-computing.com/pdfs/Altera.Fall03.pdf

[197] E. Farquhar, C. Gordon, and P. Hasler, “A field programmable neural array,” in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, May 2006, pp. 4 pp.–4117.

[198] A. Sengupta, A. Banerjee, and K. Roy. (2015) Hybrid spintronic-cmos spiking neural network with on-chip learning: Devices, circuits and systems. accessed 26.02.2016. [Online]. Available: http://arxiv.org/pdf/1510.00432.pdf

[199] spiNNaker. (2016) spinnaker project. accessed 26.02.2016. [Online]. Available: http://apt.cs.manchester.ac.uk/projects/SpiNNaker/

[200] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, “Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations,” *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[201] IBM. (2015) Introducing a brain-inspired computer - truenorth’s neurons to revolutionize system architecture. accessed 26.02.2016. [Online]. Available: http://www.research.ibm.com/articles/brain-chip.shtml

[202] A. Amir, P. Datta, W. P. Risk, A. S. Cassidy, J. A. Kusnitz, S. K. Esser, A. Andreopoulos, T. M. Wong, M. Flickner, R. Alvarez-Icaza, E. McQuinn, B. Shaw, N. Pass, and D. S. Modha. (2015) Cognitive computing programming paradigm: A corelet language for composing networks of neurosynaptic cores. accessed 26.02.2016. [Online]. Available: https://dl.dropboxusercontent.com/u/91714474/Papers/020.IJCNN2013.Corelet.pdf?cm_mc_uid=24791255049214564895444&cm_mc_sid_50200000=1456489544

[203] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” in *Handheld and ubiquitous computing*. Springer, 1999, pp. 304–307.

[204] P. Horn, “Autonomic computing: Ibm\’s perspective on the state of information technology,” 2001.

[205] X. Zhang, M. Clark, K. Rattan, and J. Muse, “Controller verification in adaptive learning systems towards trusted autonomy,” in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM, 2015, pp. 31–40.