2

Computational Intelligence in the Context of Industry 4.0

Alexander Hošovský, Ján Pitel, Monika Trojanová,
and Kamil Židek

2.1 Introduction

Artificial Intelligence (AI) has been in the forefront of research interest for relatively long time. Since its very beginning, the expectations about possible achievements had been quite high. Despite the fact that even today we are a long way from achieving human capabilities in solving

A. Hošovský · J. Pitel · M. Trojanová (✉) · K. Židek
Department of Industrial Engineering and Informatics,
Faculty of Manufacturing Technologies With a Seat in Prešov,
Technical University of Košice, Prešov, Slovakia
e-mail: monika.trojanova@tuke.sk

A. Hošovský
e-mail: alexander.hosovsky@tuke.sk

J. Pitel
e-mail: jan.pitel@tuke.sk

K. Židek
e-mail: kamil.zidek@tuke.sk

any kind of problems with similar success, in many particular areas these capabilities have already been surpassed. As pointed out clearly in Fulcher (2008), there is not a single, universally approved definition of what (artificial) intelligence actually is. In one of the broadest sense, it can be seen as “the study of making computers or programs to mimic thought processes, like reasoning and learning” (Munakata 2008). To help with this general definition, one can list some of the commonly accepted characteristics and/or capabilities that an intelligent system should have (Karray and Silva 2004):

- ability to deal with unfamiliar situations,
- learning and knowledge acquisition,
- ability to infer from incomplete or approximate information,
- sensory perception,
- pattern recognition,
- inductive reasoning,
- common sense or emotions.

As a result of intense research in this field, we have been able to achieve significant advancements in all of these points—possibly the last two are lagging behind the others, but from an industrial viewpoint, they are the easiest to overlook. At least two fundamentally different approaches have formed in the area: **symbolic AI** and **subsymbolic AI** (Munakata 2008). Symbolic AI works on a higher level of abstraction and is sometimes considered to be a traditional AI (Fulcher 2008). It tries to mimic our way of thinking using logic, reasoning, symbols, and models. The subsymbolic AI works differently and it is this part that started to take the dominant position in the field since the 1980s. It is instead more inspired by Nature itself or lower level of functioning in a human body in providing solutions to various kinds of problems—some of the best examples are nervous or immune systems or evolution and/or genetics. What is even more important from technical point of view is that subsymbolic AI is primarily data-based, making it very suitable for current implementations of intelligent systems where enormous amounts of data are generated, collected, and analyzed. Usually, a term **Computational Intelligence (CI)** is applied to the collection of methods

in subsymbolic AI. Another term sometimes considered equivalent and coined by Lotfi Zadeh (father of fuzzy logic) is **Soft Computing (SC)**. Without any attempts to contribute to academic debates regarding what exactly constitutes CI and SC and if it is indeed equivalent, suffice it to say that the three basic paradigms are considered major building blocks of CI or SC: *neural networks, fuzzy logic and evolutionary computation* (to be described later in the text).

An interesting attempt at CI classification can be found in Sumathi et al. (2018), where the four main areas of computational intelligence are distinguished: *machine learning and connectionist systems, global search and optimization algorithms, approximate reasoning, and conditioning approximate reasoning* (Fig. 2.1 right). Using this classification, the major three pillars of CI (neural networks, evolutionary computation, and fuzzy logic) could be put into the first three areas respectively. The category of GSOA (Growing Self-Organizing Array) can be tricky to classify since quite a number of various nature-inspired algorithms has been already introduced see, e.g., Xing and Gao (2014), but most of them are not well-established and the approach of searching for biological inspiration in developing new algorithms may be a bit counterproductive if the underlying mathematical foundations are disregarded (Lones 2014). Conditioning approximate reasoning category includes methods like hidden Markov models, Bayesian belief networks, or graphic models (Sumathi et al. 2018).

An important aspect and advantage over traditional AI can be observed from Fig. 2.1. It is the possibility to trade the quality of solution for lower computational load expressed through the availability

Fig. 2.1 Performance versus resources in AI and CI (P—performance, R—resources) and major CI classification

of computational resources (Fig. 2.1) (Fulcher 2008). In contrast to traditional AI, where high-quality solutions are sought at the expense of heavy computational burden, CI allows us to obtain solutions of possibly lower quality but with the reduced requirements for resources. This is significant since in many cases (e.g. NP-hard problems—Non-deterministic Polynomial hard problems) even good and not necessarily optimal solution may be acceptable.

It is now important to emphasize how the CI relates to Industry 4.0. The term Industry 4.0 is now ubiquitous as far as the area of manufacturing is concerned. It is quite understandable as its central aspect is “Smart Manufacturing for the Future” (Demir et al. 2019). As stated in the paper, its main objective is to increase productivity and achieve mass production using innovative technology. Industry 4.0 relies on several key concepts like Internet of Things, big data, cyber-physical systems, and others (Dilberoglu et al. 2017). As a result, massive amounts of data are exchanged between multitudes of devices and this very fact makes the use of a data-driven approach like CI obvious. Equipped with a plethora of powerful paradigms, CI allows any of the key concepts mentioned above to be endowed with many of the characteristics of intelligent systems.

In an attempt to emphasize the strong link between AI in general and the concept of smart manufacturing under Industry 4.0, a new term—Industrial AI (IAI)—was coined (Lee et al. 2018). Even though the authors stress its infancy in terms of clearly defined structure and methodologies, they at least set the key elements in IAI denoted with “ABCDE”: Analytics technology, big data technology, cloud or cyber-technology, domain knowhow, and evidence. The first three are well-known in the context of Industry 4.0, but domain knowhow and evidence are also considered very important for the development of IAI Ecosystem (Lee et al. 2018). Domain know-how is concerned with the knowledge of both the problem addressed by IAI and the system with its parameters and their effect on its performance.

In this short review, we concentrate mainly on the three major pillars of computational intelligence, i.e., neural network—NN, fuzzy logic—FL, and evolutionary computing. In addition to that, swarm intelligence and artificial immune systems approaches are also included since these

are also well-established methods in the field and their potential for Industry 4.0-based systems and cybersecurity is promising. Separate sections than contain some of the key concepts of Industry 4.0 (big data, cyber-physical systems), where the latest research in the field of CI use in these concepts is highlighted. The sections start with a short introduction to a given topic followed by the literature survey of the most recent research in CI-based approaches for Industry 4.0.

Figure 2.2 shows the basic structure of the chapter together with all the links between its sections. We used a classification similar to Sumathi et al. (2018) but we limited the range to three major classes—machine learning and connectionist systems, global search and optimization algorithms, and approximate reasoning (classification level). Within these three major classes, we used the division to several basic CI paradigms, with two paradigms for machine learning and connectionist systems, three paradigms for global search and optimization algorithms, and one for approximate reasoning. Therefore, the paradigm level contains six parts (deep learning, neural networks, evolutionary computation, swarm intelligence, artificial immune systems, and fuzzy logic and fuzzy systems), each corresponding to a single section within the text. In addition to the basic CI paradigms at Paradigm Level, three more sections were added to extend the focus of the chapter. The concepts of big data and cyber-physical systems are of crucial importance in the framework of Industry 4.0 and are also likely candidates to benefit from the use of computational intelligence. These can be found in separate sections

Fig. 2.2 The basic structure of the chapter

(Sects. 2.7 and 2.9) at the application level and since almost any CI paradigm can be used within their context, they are linked to all sections at the paradigm level. In order to further increase the application value of the chapter, a case study of using convolutional neural network (CNN) in object recognition during the assembly process was included as the last chapter.

2.2 Neural Networks

2.2.1 Fundamentals of Neural Networks

Some two to three decades ago the tasks of pattern recognition or obstacle avoidance were typical examples in which humans definitely excelled over the computers. Their way of processing, e.g., visual information, the capability of learning as well as performing their tasks in an unknown environment has been a source of inspiration for a longer time. Our brain relies on the parallel activity of a huge number of nerve cells called *neurons* (Fig. 2.3). This basic architecture—the interconnection of a large number of computational elements—became the idea behind artificial neural networks (simply known as neural networks—NNs). Needless to say, this inspiration is very loose and extremely simplified

Fig. 2.3 Neural network inspiration and basic operation

compared to its biological counterpart, but still proved remarkably effective for solving various kinds of problems. The main property that is of interest is **learning**—i.e., acquiring knowledge and using this knowledge to infer the right decisions for unknown situations (Karray and Silva 2004). In the case of the NNs, this learning is known as *numerical learning*—the capability of adjusting its parameters (*synaptic weights*) in response to the training signals. Basically, three types of learning are distinguished: **supervised learning** (correct answers are known), **unsupervised learning** (finding patterns in the data), and **reinforcement learning** (it is known if the answer is correct or not).

The neurons are organized in layers and their particular organization within a given network determines their *topology*. If connections are allowed only in one direction, this is known as *feedforward topology*, if feedback connections are present, this is known as *recurrent topology*. The power of nonlinear capabilities of NNs lies in their *activation function*, typically of sigmoid or hyperbolic tangent type in classical (shallow) architectures. Some typical neural network models include (Karray and Silva 2004):

- **Multi-Layer Perceptrons (MLP)**,
- **Radial Basis Function Networks (RBFN)**,
- **Kohonen's Self-Organizing Networks (KSON)**,
- **Hopfield Networks (HN)**.

MLPs are one of the most widespread classical neural network models with feedforward architecture and typically three layers (input, hidden, and output). This class of NNs could be used either for regression or classification tasks (Haykin 2009). RBFNs are a special class of feedforward NNs inspired by the biological receptive fields of the cerebral cortex (Karray and Silva 2004) mainly developed for nonlinear function approximation tasks. In contrast to MLPs and RBFNs, KHONs are typical unsupervised neural networks, where the parameters are updated without the knowledge of correct answers (Haykin 2009). They produce a low-dimension representation of the input space by retaining the original ordering (Karray and Silva 2004). The Hopfield NNs are a special class of networks with a recurrent topology that is primarily intended

as content-addressable memories with a number of locally stable states (Haykin 2009).

The performance of any of these networks is (besides other factors) highly dependent also on used training algorithm. The most famous of all is the backpropagation training algorithm typically used for MLP-like networks (Kim 2017). This is a gradient-based technique where the errors in a network are propagated backward. In the case of radial basis function neural networks (RBFNNs), it is typical to use least-squares to determine the weights once the locations of node centers as well as the widths of their RBFs are known (Liu 2013). Other methods like competitive “winner takes all” strategy or Hebbian learning rule are also possible for KSONs and Hopfield networks (Karray and Silva 2004).

2.2.2 Use of Neural Networks in the Context of Industry 4.0

The networks like MLP, RBFNN, KSON, or HNs are now considered classical types of networks that experienced a boom mostly around 2000. Since then, also thanks to the significant advancements in computer hardware, deep architectures (Section 2.8 Deep Learning) started to dominate the field due to their powerful capabilities, mainly in object and voice recognition areas but also others. With its almost implicit reliance on huge amounts of data, this fact is even more pronounced in Industry 4.0 concept. However, they still hold potential for specific applications in particular fields, especially when hybridized with other CI paradigms or when, for some reason, limited data is available.

In one of the more recent works, Yang et al. (2019) used online learning RBFNN to compensate for the unmodeled effects of the system. Together with an accurate inverse kinematic model, this network was used for a disturbance observer design. The authors used this approach for 3-PRR (Prismatic-Revolute-Revolute) compliant parallel manipulator with variable thickness flexure pivots. The use of compliant mechanisms is in line with the current trends in robotics to be used in smart factories, where the human-machine interaction is of crucial importance. A very interesting application of RBFNNs in the food industry can be

found in Shi et al. (2019), where the researchers developed RBFNN for estimating freshness of fish fillets under non-isothermal conditions. To achieve this, they selected nine optimal wavelengths from hyper-spectral imaging based on successive projections algorithms to monitor important freshness parameters.

Automated Guidance Vehicles (AGV) are considered an important part of a smart factory concept to provide higher flexibility in manufacturing and they are used for transporting goods or materials to various parts of a factory (Mehami et al. 2018). Wong and Yu (2019) used an optimization algorithm to minimize path following error based on Lyapunov direct method controller with RBF neural network estimator. This solution was proposed to address the problems of vision-based simultaneous localization and mapping when disturbances are occurring.

Optimal operation of power systems is also a significant factor in modern factories and achieving this optimality is becoming more difficult in view of the stringent requirements assumed by Industry 4.0 concept. This particular problem was addressed in Veerasamy et al. (2020), where authors used a new approach for solving non-linear transcendental power flow equations using Runge–Kutta-based modified Hopfield neural network. This was compared to the conventionally used Newton–Raphson method and showed a lower computational load with highly accurate results. Similarly, Djedidi and Djeziri in (2020) developed a new type of power estimator for ARM-based (Advanced RISC Machine) embedded systems with granularity at the level of components. This estimator was based on nonlinear autoregressive with eXogenous input (NARX) neural network and authors were able to achieve mean absolute percentage error (MAPE) of 2.2%. The results are important for IoT area, where the power consumption of embedded systems is of crucial importance.

2.3 Fuzzy Systems

2.3.1 Fundamentals of Fuzzy Systems

It is a well-known fact that the usefulness of binary logic that is fundamental in our computers is severely compromised when applied to the possible explanation of human thinking. Our ways of communication are in stark contrast with crisp and rigorous expressions needed for the proper functioning of computers. However, we are capable of solving complex problems as well as processing incomplete, uncertain, and contradicting information. This served as a powerful source of inspiration for the father of fuzzy logic Lotfi Zadeh, who introduced the concept with his seminal paper in 1965.

While the values of binary logic are restricted to 0 and 1, the fuzzy logic is multivalued, and given input may belong to a given set with any membership function value between 0 and 1. In addition to that, the particular input may also belong to more than one set with different values of the membership function. Using this concept it is possible to express the degree of truth (Antão 2017), which is defined for certain types of membership functions assigned with linguistic labels. These are known as fuzzy sets and may be assigned labels like “*low*”, “*high*”, “*very low*”, “*very high*” and similar. These fuzzy sets have specific shapes, with triangular, trapezoid, or Gaussian being the most common. Besides the process of fuzzification of crisp values, it is also the use of *If–Then* rules in fuzzy systems that make it possible to mimic the way of human reasoning when, e.g., controlling systems or processes. Using the definition of such rules, it is possible to incorporate the expert’s knowledge in certain areas into a fuzzy system (Pedrycz and Gomide 2007). *If* parts of the rules are known as *antecedents* and *Then* parts as *consequents* and based on the form of consequents two most common types of fuzzy systems are distinguished:

1. **Mamdani**
2. **Takagi–Sugeno**

Fig. 2.4 Fuzzy logic inspiration and basic operation

The first type uses fuzzy sets in consequent parts of the rules while the second one uses a linear function of its inputs. In general, every fuzzy system contains four basic parts: *fuzzifier*, *rule base*, *inference system*, and *defuzzifier* (Antão 2017). The purpose of a fuzzifier is to convert the crisp value of a given variable into a fuzzy domain. The rule base stores the knowledge in the form of *If-Then* rules (Fig. 2.4) which can be extracted from an expert or numerical data. Using the inference engine, it is possible to make basic algebraic manipulations with fuzzy sets, while defuzzifier calculates the crisp value at the output of a fuzzy system based on the aggregated results from all active rules.

It was recognized very soon that even though fuzzy systems were designed for handling the uncertainty in data, once all parameters are determined it becomes completely certain (Antão 2017). To address this, Type 2 fuzzy systems were introduced where the value of membership function is uncertain and specified also using a fuzzy set with a value from $[0,1]$ interval. All possible values of membership functions are then limited from above with *upper membership function* and from below with *lower membership function* and the area between them is known as a *Footprint of Uncertainty (FOU)*.

2.3.2 Use of Fuzzy Systems in the Context of Industry 4.0

Fuzzy logic is an important CI paradigm, which belongs to the category of approximate reasoning methods (Fig. 2.1). In contrast to other purely data-driven CI methods, fuzzy logic (FL) stands on the boundary of traditional AI, of which expert systems are a prime example, and subsymbolic, data-based methods. As a result of this, they have certain advantages in being able to incorporate the expert knowledge in their structure, while also taking care of uncertainty and imprecision of this knowledge.

With advances in Industry 4.0 concept implementation in the manufacturing process, the reality of smart manufacturing becomes imminent. In that case, the interconnection of many elements that share enormous amounts of data is one of the central points to consider when designing the control part of this network. Researchers in Huo et al. (2020) proposed to use a fuzzy control system to provide real-time analysis of information on an assembly line. To improve the performance, two types of a fuzzy controller were used: one of them of Type 1 and the second one of Type 2. The former handled the situations where the need for re-balancing the assembly line for satisfying demands was decided. The latter one's purpose was to adjust the production rate in order to eliminate blockages and increase the utilization of machines. More specifically, authors in Lu and Liu (2018) tried to address the issue of keeping the quality of a manufactured product within acceptable bounds based on Taguchi methods. For this, they developed a fuzzy nonlinear programming model based on a fuzzy signal-to-noise ratio. By using this approach, they were able to obtain optimal solutions of lower and upper bound fuzzy S/N (Signal/Noise) ratio.

As one of the principal technologies under Industry 4.0, Internet of Things may certainly benefit from the application of fuzzy logic on many levels. One of them is the use of Wireless Sensor Networks (WSNs) used for sensing the environment and collection/sending of data to the base station for analysis (Thangaramya et al. 2019). In this area, both the intelligent routing and energy optimization are aspects that need to be addressed to keep quality of service in the network at an

acceptable level. The authors proposed the use of neuro-fuzzy rule-based cluster formation and the routing protocol to handle these issues and also employed a convolutional neural network for rule formation on discovering energy-efficient routing. Fuzzy logic-based methods can also help with the processing of vast amounts of data generated by a large number of interconnected devices. In Bu (2018), authors propose a high-order tensor fuzzy c-mean algorithm, which was said to achieve much higher clustering efficiency compared to a traditional algorithm.

With the vast amounts of data becoming ubiquitous, the concept of big data and methods for its efficient handling is central to Industry 4.0. The principles of fuzzy logic hold great potential for applications in big data analytics. In Shukla et al. (2020), researchers proposed the use of interval type-2 fuzzy sets for handling the veracity issue in big data to prevent the unusability of data. The problem of handling big data was addressed also in Zhang et al. (2020b), where a quantitative model and method based on fuzzy DEcision MAKing Trial and Evaluation Laboratory (DEMATEL) were proposed. As reported in the paper, it could be used as a theoretical basis for handling big data by industry or government. Likewise Chen et al. (2020) used DEMATEL for determining the criteria weights in a smart supply chain. However, the authors identified problems simultaneous manipulation of internal and external uncertainties. For this, they proposed a hybrid rough-fuzzy DEMATEL-TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) approach for sustainable supplier selection in a smart supply chain.

2.4 Evolutionary Computation

2.4.1 Fundamentals of Evolutionary Computation

Evolutionary computation is a paradigm that is widely regarded as one of the main pillars in the field of computational intelligence. Taking its inspiration from neo-Darwinism, this collection of computational methods makes use of basic principles of evolutionary biology, natural selection process, and genetic variations (Castro 2006). The genetic variations happen at the level of chromosomes with their basic functional

units named *genes*. These form the genetic makeup of an individual termed *genotype*. This makeup affects the observable characteristics or *traits* of an organism. It is through these traits that an individual can show its better adaptation to the environment and thus increase the probability of survival and reproduction—this is known as the *fitness* of given individual (Fulcher 2008).

It was natural to adapt these principles to a computational form, where it is possible to search for the solutions to optimization problems. These methods were collectively named as *evolutionary computation* and include three different approaches (Castro 2006):

- **genetic algorithms,**
- **evolution strategies,**
- **evolutionary programming.**

In general, evolutionary algorithms maintain a population of individuals which themselves represent solutions to the problem with various forms of encoding. In analogy with the main inspiration, the individuals within the population are evaluated according to their fitness—i.e., how well they are adapted to their environment or, in terms of optimization problem solving what is the value of an objective function. From this population, a certain number of individuals is selected to mating pool, where the *crossover* (recombination of their genetic information), as well as mutation (alteration of existing genetic information), take place, with all these processes corresponding to one *generation* (Fig. 2.5).

The genetic Algorithm (GA) is one of the most widespread evolutionary algorithms. In its well-known form, it uses solutions encoded as binary numbers (bitstrings), which are usually known as *chromosomes* (Sumathi et al. 2008). Each place in this chromosome is known as *locus* and its possible value at this position is called an *allele*. The whole chromosome then represents one solution to the problem at hand and its quality is evaluated using the fitness function. The parents for mating can be selected using *roulette wheel selection* or *tournament*. The form of crossover operation is in the simplest case one- or two-point, which means the number of positions at which the chromosomes are cut and recombined. In binary genetic algorithm (GA), the mutation can be

Fig. 2.5 Evolutionary computation inspiration and basic operation

carried out as simple flipping of the original value (from 0 to 1 or vice versa). This has to be done with small probability so that the valuable genetic information contained in the population is not destroyed by excessive random modifications.

Since the introduction of GA, a very high number of variations have been developed to address various aspects of the original implementation. This includes messy GA (Goldberg et al. 1995), island GA (Cantú-Paz 1998), niching GA capable of locating multiple solutions (dynamic niche sharing [Miller and Shaw 1996], nondominated sort GA [Srinivas and Deb 1991]), coevolutionary shared niching (Goldberg and Wang 1997) and many others.

Differential Evolution (DE) is an evolutionary algorithm that basically differs from the genetic algorithm in that instead of crossover, the mutation is applied first to generate the so-called *trial vector*. Only after this step, the crossover operator is applied to produce one offspring. In addition to that mutation step sizes are not sampled from a prior known probability distribution function (Engelbrecht 2007). The working principle of differential evolution is based on the concept of *difference vectors*, which correspond to the magnitudes of distances between individuals in the population. If those distances are large (individuals are far away from each other), the search space should be explored (taking large steps). However, if the opposite is true, it is reasonable to exploit the search

space and look for the solutions only in the close vicinity of the current position (Feoktistov 2006). Therefore, the mutation steps are calculated as weighted differences between individuals that are selected in random (Engelbrecht 2007).

In addition to classic variations of the basic differential evolution (DE) algorithm denoted with DE/ $x/y/z$, where x is the method of target vector selection, y is the number of difference vectors and z is the method of crossover, many other modifications have also been introduced. These modifications include gradient-based hybrid DE (Chiou and Wang 1998), DE-hybridized GA (Hrstka and Kučerová 2004), DE-hybridized particle swarm optimization (PSO) (Hendtlass 2001), dynamic self-adaptive DE (Chang and Xu 2000), angle modulated differential evolution (Pampara et al. 2006) and others.

Coevolution is a special type of evolution, where the complementary interaction between species is considered (Engelbrecht 2007). This can happen, e.g., in *predator-prey* interaction, where one of the species evolves to be better in escaping a predator while the other one evolves to be better in catching this prey. That is, this interaction is complementary because the failure of one of the species naturally means the success of the other one. The main difference compared to the standard evolution-based algorithm is that in coevolution type algorithms one does not use the absolute fitness function to evaluate the optimality but attempts to achieve optimality through defeating opponent Engelbrecht (2007). There are basically two types of coevolution:

- **competitive,**
- **cooperative.**

The predator-prey model of coevolution can be considered a competitive type since, as mentioned above, the success of one of the species leads to the failure of the other one. In contrast, a cooperative type of coevolution involves the possible improvement of both or one of the species.

2.4.2 Use of Evolutionary Computation in the Context of Industry 4.0

Evolutionary computation is a well-established CI paradigm, which, either in its original or hybridized form, has been used successfully in many applications. Due to its population character, it lends itself to a parallel implementation to make it more effective. Its possible use in Industry 4.0 concept is manifold—if the problem at hand can be cast as an optimization problem, evolution-based algorithms can be used for a search of the solution. These problems can range from the controller design and/or neural networks training to job-shop scheduling and supply chain optimization.

The need to apply advanced computational methods in the area of logistics and supply chain management as a part of smart manufacturing under the concept of Industry 4.0 is evident. In this scenario, the problem of resource-constrained job scheduling is an important one and bio-inspired computational methods are often applied to address it. The researchers in Nguyen et al. (2019) used a hybrid optimization method based on differential evolution, iterated greedy search, mixed integer programming as well as parallel computing to solve the problem of resource-constrained job scheduling for large-scale instances. The problem of supply chains was tackled also in Saif-Eddine et al. (2019), where specifically the total supply chain cost was optimized. Since this belongs to the group of NP-hard problems, an improved genetic algorithm was designed and used to address the problem. It was shown that this modification outperformed classical GA for two instances (10 and 30 customers).

Cognitive Radio Networks (CRN) are an important type of networks with applications ranging from wireless sensor networks to Medical Body Area Networks, and are thus an important part of communication framework within Industry 4.0. In CRN the energy efficiency issue is of utmost importance, which is addressed in Tang and Xin (2016) through the use of new energy efficiency metric. The optimization problem itself is solved using a chaotic particle swarm algorithm and coevolution methodology, which helps to decompose the original problem into several smaller ones.

The concept of distributed manufacturing is of great interest in meeting current demands on quick responses to the market changes and the sharing of resources. On the other hand, using this concept requires to address the problem of job assignment to different shops as well as its sequencing. The researchers in Zheng et al. (2020) used a cooperative coevolution algorithm for multi-objective fuzzy distributed hybrid flow shop. The coevolution part of the algorithm is proposed to achieve a proper balance between the exploration and exploitation capabilities based on the information entropy and elite solutions diversity.

Additive manufacturing is considered to be a crucial part of Industry 4.0-based manufacturing, which assumes the integration of intelligent production systems and advanced information technologies (Dilberoglu et al. 2017). Following this, Mele and Campana (2020) used evolutionary computing for addressing the problem of parts build orientation, based on the life-cycle impact assessment indicators used for modeling the Pareto front of environmentally non-dominated solutions. Likewise, Ewald and colleagues (Ewald et al. 2018) adapted evolutionary algorithm for varying the size, orientation, and position of wrought material in hybrid manufacturing strategy that combined laser metal deposition and milling or turning.

2.5 Swarm Intelligence

2.5.1 Fundamentals of Swarm Intelligence

Swarm intelligence has been a subject of interest among researchers in technical fields almost since the introduction of this term. Even before that, the behavior of the collection of certain animals was found intriguing and could serve as a remarkable source of inspiration Fig. 2.6. As a matter of fact, many systems in the industry can be viewed as a collection of simple agents cooperating among themselves in the same environment. In this regard, swarm intelligence can be defined as “the emergent collective intelligence of groups of simple agents” (Bonabeau et al. 1999; Tan et al. 2010; Nayyar et al. 2018). By observing the behavior of those simple agents in nature (be it birds, ants, bees, or

Fig. 2.6 Swarm intelligence inspiration and basic operation

others), we see the emergence of properties which are not inherent in any of those individuals. To describe two fundamental properties of swarm intelligence, one can refer to two different animal species—ants and birds. In both species, the self-organization as one of those properties can be observed. According to Blum and Merkle (2008), self-organization is “a process in which patterns at the global level of a system emerge solely from numerous interactions among the lower-level components of the system”. In addition to that, the behavior of ants includes another fundamental property, which is a division of labor viewed as parallel execution of different tasks by agents in a swarm (Nayyar et al. 2018).

While many algorithms inspired by the behavior of swarms have been developed, two of them form the backbone of so-called swarm intelligence algorithms, i.e., particle swarm optimization and ant colony optimization (Engelbrecht 2007).

Particle Swarm Optimization (PSO) is a population-based search algorithm inspired by the behavior of birds when flying in a flock (Engelbrecht 2007; Yang et al. 2013). In the form of an algorithm, the number of particles (members of the population) flies through hyperdimensional search space in an attempt to find an extremum of a given function, possibly under certain constraints. The position of each of the particles is changed based on its own experience as well as the experience of its neighbors through the cognitive and the social components (Engelbrecht 2007).

Ant Colony Optimization (ACO) is also a population-based algorithm inspired by the foraging behavior of real ants. The problem solved by the algorithm can be cast as a problem of finding the shortest path between two nodes, which is achieved through so-called *stigmergy*. Stigmergy is an “indirect communication mediated by numeric modifications of environmental states which are only locally accessible by the communicating agents” (Dorigo and Di Caro 1999). The ants choose paths in a probabilistic manner in response to the amount of pheromone concentration on a given path.

From the inception of both types of algorithms many modifications have been proposed including social-based PSO (Messerschmidt and Engelbrecht 2004), GA-PSO hybrid algorithm (Angeline 1998), NichePSO (Agrafiotis and Cedeño 2002), craziness PSO (Kennedy and Eberhart 1995), quantum-behaved PSO (Fang et al. 2010), ant colony system algorithm (Ippolito et al. 2004), max–min AS (Stützle and Hoos 2000), Ant-Q algorithm (Mariano and Morales 1999), Antabu (Fonlupt et al. 2006) and many more. All these modifications addressed various problems of the original algorithms either in general or in specific applications, where they contributed to the improved performance.

2.5.2 Use of Swarm Intelligence in the Context of Industry 4.0

Similar to evolutionary computing, swarm intelligence-based algorithms have become quite popular as non-gradient optimization techniques applied to hard optimization problems. The basic PSO is quite simple but further modifications increased its complexity and improved its performance. The newer implementations like quantum-behaved PSO are quite powerful, even for high-dimensional problems where other optimization techniques might fail. The cooperation of simple agents as seen in swarm intelligence has not been used solely in optimization algorithms but served as an inspiration for many other approaches proposed for the smart manufacturing area and/or Industry 4.0 in general.

In Sun et al. (2018), researchers used swarm intelligence for community detection, which is a task of critical value in the analysis of complex

networks. This is especially important for dynamic networks, where the properties of decentralized, self-organized, and self-evolving systems are of importance. The use of swarm intelligence also addresses the problem of overlapping community detection since it can handle the joining of the vertex into multiple communities and also the addition or deletion of a vertex dynamically. In particular, a particle swarm optimization algorithm was used in Gill et al. (2018) for the problem of cloud resource scheduling, which requires the mapping of cloud resources to cloud workloads. By using PSO, the parameters of Quality of Service (execution cost, time, and energy consumption in particular but also others) could be significantly reduced. Taking into account the specific properties of the ACO algorithm and its variants, it is natural to consider it for solving the problems of routing as a part of Internet of Things implementations. This was used in Thapar and Batra (2018), where the network of sensor nodes was seen as a colony of ants. RPL protocol (Routing Protocol for Low-power and lossy networks) was used to build a destination-oriented directed acyclic graph using the objective function, which was responsible for fixation of the rank of node and selection of best directed acyclic graph using ACO. This algorithm was also adapted to resource distribution optimization in Hong et al. (2019) in the form of resource indexing optimization. The velocity and position of cluster resource indexing were updated based on the ant colony trajectory with constraint condition on the minimum variance of the fitness function.

With an increasing number of various embedded devices in IoT, the scheduling tasks can be considered NP-hard problems for which no polynomial-time algorithms might be available. Therefore, the use of metaheuristics like PSO can be suitable to obtain good (or even close to optimal) solutions. Authors in Xie et al. (2019) used PSO for the problem of workflow scheduling in a cloud-edge environment. They introduced a Directional and Non-local-Convergent PSO (DNCPSO), which employed non-linear inertia weight where the selection and the mutation operations were performed using the directional search process.

The problem of path planning for mobile robots is relevant also for the concept of smart manufacturing, where the extensive use of AGVs is

expected. Depending on the conditions, this is usually a computationally demanding task, where the bio-inspired computational methods can be of great benefit. Dewang et al. (2018) used adaptive particle swarm optimization (APSO) for the path planning of a mobile robot. This was showed to be faster than using a conventional PSO algorithm.

2.6 Artificial Immune Systems

2.6.1 Fundamentals of Artificial Immune Systems

Similar to the human brain and nervous system in general, the immune system is well-known for its remarkable properties in maintaining the balance of internal state of humans, especially in response to the invasion of external harmful agents (e.g., viruses and bacteria). This system is extremely complex and sophisticated, being in constant interplay with other systems within the human body to achieve homeostasis (dynamic state of equilibrium). Even there are several layers of body protection against invaders, the most important division of the immune system in terms of its function and properties is to the *innate immune system* and *adaptive immune system*.

The innate immune system is known to be able to mount a response against harmful agents by recognizing their generic molecular patterns not present in the cells of a host but only in invading pathogens (Castro 2006). When those agents damage the cells of a host, the innate immune system provides co-stimulatory signals, needed, e.g., for the action of the adaptive immune system. Moreover, it is the innate immune system that provides a faster response to the invasion, while the adaptive system starts to act. On the other hand, it is the adaptive immune system that is capable of fighting even against the invaders never seen before. Furthermore, when these pathogens are present again, the adaptive immune system can mount a faster response through the “immune memory” (Engelbrecht 2007; Castro 2006).

It is an adaptive immune system in particular that became the main source of inspiration for developing algorithms loosely inspired by its function. According to Engelbrecht (2007), some of the capabilities

of the natural immune system usable in computational tools are the following:

- The immune system can distinguish between self and foreign/non-self cells (and knows their structure).
- Foreign cells can be *dangerous* or *non-dangerous*.
- Lymphocytes (a certain type of white blood cells) are subject to cloning and mutation to adapt to the structure of foreign cells, which leads to the formation of *memory*.
- Lymphocytes have coordination and co-stimulation among them, forming *immune networks* as a result.

An artificial immune system algorithm is a population-based algorithm that can be used for clustering and/or optimization problems. In its basic form, this algorithm uses artificial lymphocytes that form the population of the solution to a given problem. After selecting a subset of this population, the affinity (the measurement of similarity or dissimilarity) between this subset and antigen is calculated. The calculation of affinity can be applied also to ALCs (Artificial LymphoCytes) themselves in analogy to immune networks. Then, based on these results, some of the ALC can be selected (through negative or positive selection) to be cloned and mutated to find ALCs with an even better affinity with antigen. Some of them can be selected to become memory cells for the secondary response of artificial immune system (AIS) when similar antigens are encountered.

Artificial immune networks are a special type of artificial immune system model, where the main difference compared to clonal selection-based models is their characterization as dynamic systems capable of functioning also without the antigen stimulation (Castro 2006). From a mathematical point of view, it is natural to describe such systems using ordinary differential equations allowing easy incorporation of real immune system (IS) properties like learning, memory, self-tolerance, and network interactions (Castro and Timmis 2002). In this kind of network, it possible to achieve stimulation of cells by another cell or a foreign antigen, while its suppression occurs due to recognition of self only (Castro 2006).

In addition to the basic AIS algorithm, many other modifications appeared some of which were based on clonal selection theory models (like CLONALG—CLonal selection ALGORITHM (Castro and Zuben 2000), AIS with dynamic clonal selection (Kim and Bentley 2002) or multi-layered AIS (Knight and Timmis 2002)), immune network theory models (AINE—Artificial Immune NETWORK (Timmis and Neal 2001)), EAINE—Enhanced Artificial Immune NETWORK (Nasraoui et al. 2002), aiNet (Castro and Zuben 2002) or danger theory models (Aickelin and Cayzer 2002).

2.6.2 Use of Artificial Immune Systems in the Context of Industry 4.0

Artificial immune systems as a paradigm loosely inspired by the functioning of the natural immune system offer interesting capabilities like adaptability, self-learning, and robustness that can be used for various tasks in data processing, system modeling, and control, fault detection or cybersecurity. All these aspects make it a suitable paradigm for addressing the problems in the context of Industry 4.0.

An interesting approach is used in Wang et al. (2018b) for optimizing the manufacturing process through energy monitoring as well as re-scheduling of manufacturing. The researchers did not use a standard approach where the conditions of the manufacturing process are known in advance but recorded and analyzed the data of energy consumption using neural networks and statistical tools. AIS algorithm was then used to tackle the situations with highly variable conditions in the manufacturing process. Researchers in Semwal and Nair (2020) realized that the centralization approach for the implementation of networked environments in cyber-physical systems is expensive concerning the increasing number of devices and the flow of information. While decentralization and distribution of the architecture address this issue, it remains a challenge to find the best solutions for problems distributed across devices. Inspiration is therefore taken from the function of the immune system (immune networks, danger theory, and clonal selection), which works as a decentralized system with capabilities of adaptivity, self-learning,

and self-organization. The concept of AIS in the context of Industry 4.0 can be used also for addressing the issue of cybersecurity threats. This was applied in Zhang et al. (2011), where a distributed intrusion detection system in smart grids was used. The communication was provided through several wireless mesh networks with 802.15.4, 802.11, and World Interoperability for Microwave Access (WiMAX) standards, which presented cybersecurity threats. To tackle these problems, researchers proposed a distributed intrusion detection system for smart grids where several analyzing modules based on support vector machines and artificial immune system paradigms were used. These modules helped to detect and classify malicious data as well as possible cyberattacks.

The latest study of Aldhaheeri et al. (2020) provides a review of the literature and recommendations for further research in the field of AIS application to secure IoT. This work tries to fill the gaps in the coherent and systematic presentation of AIS capabilities for Internet of Things, especially in terms of cybersecurity. By offering an exhaustive survey of the past as well as recent research in the area, this paper corroborates the fact that this paradigm holds the potential for current and future applications within Industry 4.0 concept.

Autonomous mobile robots are an important part of the smart factory concept, where one of the tasks may be the transport of materials across the factory floor while avoiding obstacles and identifying pickups or material dropoff in real-time. This presents a problem of navigation in an uncertain and/or unstructured environment, where the properties of adaptivity and self-organization of natural immune systems can help. Authors in Akram and Raza (2018) proposed the concept of the Robot Immune System to maintain the robot's internal health-equilibrium (analogy to homeostasis). For this, a robot uses health indicators (e.g., energy and temperature) to detect any abnormalities in its function. This eventually leads to the state of inflammation, which activates the first innate and subsequently adaptive immune system.

The monitoring of the proper operation of systems may be critical under the concept of Industry 4.0. Should any fault occur in a system, it is not only important to detect it but also to try to locate it. These approaches are known as *Fault Detection and Isolation (FDI)* and can

be typically model-based or signal processing-based. In Costa Silva et al. (2017) authors presented a review of three different AIS approaches to FDI—the toll-like receptor algorithm, the dendritic cell algorithm, and the danger theory-based algorithm.

2.7 Big Data

The importance of data in general within Industry 4.0 cannot be stressed enough—it is simply one of the pillars upon which this concept is built. From the smallest and least complex systems to smart factories, the existence of massive amounts of data need to be taken into account and processed accordingly. While sometimes it may not be utterly clear what exactly should be considered big data and what should be not, there is some consensus regarding the main aspects according to which we could evaluate data in question. In Iqbal et al. (2020a), the five Vs of big data are presented:

1. **Volume**
2. **Velocity**
3. **Variety**
4. **Veracity**
5. **Value**

Without doubts, the *volume* should be the single most obvious aspect of big data. Enormous amounts of data are generated and stored in an instant and these volumes can reach even zettabytes for the whole Internet (Iqbal et al. 2020a). Even though amounts generated, e.g., in a smart factory as a whole will be much smaller, they still have large enough volume to necessitate a special approach for its analysis. *Velocity* refers to the speed at which this data is generated but also streamed or stored (Iqbal et al. 2020b). Since the source of data can be as diverse as social media or data from sensors, it is obvious that the structure of data can differ significantly—this falls under the category of *variety* of data. Due to the very large volumes of data generated at high velocity, a lot of noise is contained in the data and its trustworthiness may be

impaired and this aspect is known as *veracity* of data. The last V of the list named *value* is related to the meaningful insight of data associated with its usefulness in identifying important patterns (Iqbal et al. 2020a). It should be noted that in some sources only three Vs (volume, velocity, and variety) as the main characteristics of big data are given (Khan et al. 2019).

After data is collected, it rarely can be used in its raw format and has to undergo some form of pre-processing and analysis (Fig. 2.7). As variety aspect of the data implies, it can be from different sources be it log files, industrial sensors, webpages, etc. Moreover, in its raw form, it includes a lot of unimportant information as well as some redundancy which has to be tackled in some way. This helps to maintain the data consistency as well as optimize its storage requirements, which can be quite important when vast amounts of data are considered. When considered as a whole, the methods in this step allow uncertain and incomplete data to be modified and/or removed while also making the dataset free of repetitive or superfluous information (Khan et al. 2019).

It is mainly the last step (titled Big Data Analytics in Fig. 2.7), where the use of machine learning (or specifically computational intelligence) techniques may be beneficial. Machine learning as an approach, where mathematical models derived from sampled data for making predictions are used, is suitable for finding patterns in the datasets. These methods are usually statistical in their nature and the existence of models derived using them is important for the process of decision making. On the other hand, computational intelligence represents a collection of typically

Fig. 2.7 Three basic steps in big data analysis process

nature-inspired approaches to problem-solving, often offering good solutions but usually without guarantees of optimality. However, by using these approaches previously intractable problems can be addressed efficiently with acceptable results. The problem of big data analytics is nowadays encountered in many fields and quite often the results can be generalized to some extent from one field to another.

Neural networks in general are one of those CI paradigms that are used quite often in Big Data Analytics. Hernandez et al. in (2020) used two new hybrid neural architectures in which morphological neurons and perceptrons were combined. Both types were used for feature extraction and trained by a stochastic gradient optimization technique. It was shown in the paper that multi-layer neural network (MLNN) required a lower number of learning parameters than other architectures. The problem of big data analysis was addressed also in Anbarasan et al. (2020), where authors used a combination of IoT and CNN in big data scenarios with the flood detection system, giving better results than other competitive methods. In addition to feature extraction, feature selection is also one of the key points in machine learning to eliminate the redundancy of data or avoiding the curse of dimensionality. Nature-inspired metaheuristics appear particularly suitable for this kind of problems, due to their capabilities of finding good solutions for NP-hard problems. Researchers in Abdi and Feizi-Derakhshi (2020) extended Search Manager for multi-objective problems and used it for EEG signal (ElectroEncephaloGraphy) analysis with reportedly good results. Likewise, this problem (of feature selection) was also studied in Nguyen et al. (2020), where the authors reviewed various approaches to feature selection problems in big data scenarios using swarm intelligence algorithms. An ensemble of three methods (non-dominated sorting genetic algorithm, differential evolution, and multi-objective evolutionary algorithm based on dominance and decomposition) combined with CNN was used in Essiet et al. (2019) for efficient data mining from dedicated databases of big data for a gas sensor. An important problem in the field of big data analytics is that of database mining. In particular, Djenouri et al. (2019) researched the association rule mining problem for which bees swarm optimization was considered effective but too computationally

demanding. The authors developed Graphics Processing Unit (GPU)-based bees swarm optimization miner, where the GPU was used as a co-processor and found the method to be $800\times$ faster than CPU-based (Central Processing Unit) method.

2.8 Deep Learning

2.8.1 Fundamentals of Deep Learning

As was mentioned in Section 2.2 neural networks, the neural networks consist of several simple computing units called neurons organized in layers and interconnected by synaptic weights. It is through the modification of these weights that neural networks are capable of learning (whether in a supervised or unsupervised manner). The term “deep” is related to the depth of neural network architecture, which is used whenever the network contains at least two hidden layers (Fig. 2.8). Thus, in principle, the deep architectures are closely related to their shallow counterparts from a structural viewpoint. However, the increase in the number of hidden layers offers significant improvements in network

Fig. 2.8 Structure of deep feedforward neural network with k inputs, l output, o hidden layers and generally different number of neurons in each hidden layer

performance that cannot be matched by increasing the number of neurons in only a single hidden layer (Aggarwal 2018).

It is clear that the idea of including at least one additional hidden layer to a network to improve its performance is not a new one. Yet, the successes achieved by using deep neural networks are more recent. The reasons for that can be found in three main problems that were effectively solved in the last fifteen years (Kim 2017): *vanishing gradient problem*, *overfitting*, and *computational load*.

The problem of vanishing gradient refers to the drop in gradient values, which can happen when multiple layers are present. Actually, the updates in weights for earlier layers become almost negligibly small which equals the stop of the training process. The use Rectified Linear Unit (ReLU) function the derivative of which is a constant address this problem efficiently (Kim 2017). Also, by using a large number of layers with many parameters the risk of overfitting becomes much more imminent—therefore, new effective methods for prevention of overfitting were needed. A simple but very powerful method is called *dropout*, where some neurons are set to zero during the training and this encourages learning sparse representations. Moreover, the larger the network is the longer time it takes to have it trained (given the same hardware). That is why the effective training of deep neural networks was conditioned by the advances in computer hardware (GPUs are particularly suitable for this task).

Some of the well-known architectures in the area of deep learning include:

- Autoencoders,
- Recurrent neural networks,
- Convolutional neural networks.

2.8.1.1 Autoencoders

Autoencoders are special types of networks, where the input dimensionality is the same as the output dimensionality. The main point of their function is that it assumed that the number of neurons in the hidden layer is lower than the number of inputs/outputs, thus allowing for a

Fig. 2.9 Structure of deep autoencoder with four inputs/outputs and 3-2-3 neurons in three hidden layers

more compact representation of input data (Aggarwal 2018; Sengupta et al. 2020). Since data is passed through a structurally constricted part of a network, the result is information loss which can be expressed through a common error criterion (e.g., SSE—Sum of Squared Errors) (Aggarwal 2018). Autoencoders can be also trained using standard backpropagation training algorithms (Sengupta et al. 2020). The process of getting a compact representation of input using a constricted structure of an autoencoder is called *encoding* (and this part *encoder*) while the process of reconstruction of original data from the encoder is called *decoding* (and this part *decoder*). When multiple hidden layers are used in an autoencoder, it can be called *deep autoencoder*. While not necessarily so, the hidden layers of deep autoencoder are typically symmetrically structured (Fig. 2.9) (Aggarwal 2018).

2.8.1.2 Recurrent Neural Networks (RNN)

Sometimes it is important to take into account not only the relationship between the input and output data (without their explicit dependence on

each other) but also the sequential character, which is necessarily associated with time. In that case, the ordering of data in time is crucial and the concept of time-stamp where the values with successive time-stamps depend on each other (Aggarwal 2018). In that case, the use of recurrent neural networks might be beneficial—in contrast to feedforward neural network (FNNs), their architecture includes some kind of looping (in simplest case self-loops associated with the hidden state of neurons may be present). RNNs hold great potential for modeling processes and systems with complex nonlinear dynamics, with long short-term memory networks being one of the well-known and powerful types.

Long short-term memory (LSTM) networks are a special type of recurrent neural networks (Fig. 2.10), which uses a different architecture compared to standard RNNs like Elman or Jordan. These networks contain three gates—forget gate, input gate, and output gate—which provide fine-grain control over data written into long-term memory (Sengupta et al. 2020). The training of RNNs is known to be difficult mainly due to the issues with vanishing and exploding gradients as well as highly varying sensitivities of the error surface to different temporal layers (Aggarwal 2018). Since weights are shared and these networks can become very deep after unfolding in time, successive multiplication with weights smaller than 1 in gradient calculation tends to zero (vanishing gradient) while for weights larger than 1 it tends to very large values (exploding gradient). This issue is addressed by using the above-mentioned three gates, where the forget gate controls the amount of information to be removed from previous cell state c_{t-1} whereas input

Fig. 2.10 Basic structure of a long short-term memory network

gate decides with what amount of information contained in cell state candidate $\tilde{\mathbf{c}}_t$ should a new cell state $\mathbf{c}_t$ be updated (Bianchi et al. 2017). The output gate then selects which part of the cell state would be returned as output. Four nonlinearities in total are used in LSTM structure, two of them are placed in the input gate (hyperbolic tangent function and sigmoid function) and one is placed in both forget gate and output gate.

2.8.1.3 Convolutional Neural Networks (CNN)

Convolutional neural networks belong to a special type of neural network particularly suited for the tasks of image recognition (Fig. 2.11). Their architecture is quite different compared to previously mentioned neural networks and is inspired by how images are processed in the visual cortex of the brain (Goodfellow et al. 2016; Kim 2017). It is only during the last years that their potential started to be explored and used in the area of machine vision with triumphant dominance over other techniques. Previous approaches were based on methods demanding extreme expenditure in terms of cost and time for development and offering inconsistent performance (Kim 2017; Goodfellow et al. 2016). The reason for that lies in the need for feature extractor design, which could be specific for a given application and therefore lacking the properties of the general-purpose image recognition tool. This issue is specifically addressed in CNNs, where the design of feature extractor is a part of the training process and can thus be used generally (Kim 2017; Sengupta

Fig. 2.11 Basic structure of a convolutional neural network

et al. 2020). The *feature extraction network* consists basically of two types of layers: **convolutional layer** and **pooling layer** (Goodfellow et al. 2016).

The function of convolutional layers is based on a mathematical operation called (quite expectedly) *convolution*. This is performed in 2D and acts as a set of digital filters (Kim 2017) to produce so-called feature maps, where some of the features of the original image are enhanced. The pooling layers then serve as dimension reduction elements by merging neighboring pixels into one based on either max or averaging operation. The last part of CNN structure is a *classification network*, which is typically represented by the fully connected network with a number of outputs corresponding to the number of classification classes (Sengupta et al. 2020).

2.8.2 Use of Deep Learning in the Context of Industry 4.0

Deep learning is one of the most perspective CI paradigms for the concept of Industry 4.0 as a whole. Recent advancements in this field confirmed its usefulness for a wide spectrum of problems. It is well-established that their success depends on the availability of huge amounts of data as well as high-performance hardware. Provided these requirements are met, the capabilities of DL in solving certain types of problems can even surpass that of humans.

Considering the crucial position of cloud and fog computing in Industry 4.0 framework, the risk of cyberattacks with potentially disastrous effects is very high. Therefore, the importance of cybersecurity has become paramount and a multitude of approaches have been proposed to address this issue. Many of the proposals make use of AI techniques, which help to achieve high performance under highly variable conditions of the operation of real-world computer system resources. In Almiani et al. (2020), authors proposed the use of the deep recurrent neural network for fog computing security, where the effectiveness of its use was demonstrated using various metrics including Mathew correlation and Cohen's Kappa coefficients.

Although powerful in the applications where a limited amount of data is available for characterizing the properties of various systems, more typical paradigms of computational intelligence like shallow neural networks, support vector machines, logistic regression, etc. may have limited performance under the assumption of massive amounts of data. This assumption is of crucial importance in, e.g., smart manufacturing, where the use of deep architecture models may be beneficial. This topic is researched in Wang et al. (2018a), where an extensive treatise of the methods and applications of deep learning in the field of smart manufacturing is presented. The advantageous use of deep learning methods within the concept of smart manufacturing can happen on many different levels. A good example of this is presented in Andersen et al. (2019), where deep reinforcement learning is used for industrial robots to cope with natural variations in the brine injection process during the production of a meat product. The prospect of deep learning application in the field of robotics is further emphasized in Wang et al. (2020a) where it is used in a multi-robot scenario. In this work, a multi-robot cooperative algorithm using deep reinforcement learning is designed based on the use of Duel network structure, where two streams representing the state value function and state-dependent action advantage function appear and their results are merged.

In particular, the processes in manufacturing themselves may benefit significantly from applying deep learning concepts—be it for their analysis or, quite typically for current trends, or visual inspection. Researchers in Wang et al. (2020b) prepared a tutorial for researchers on how to apply (and also understand) deep learning in manufacturing, with welding used as an example. They discussed two of the most typical techniques, namely the convolutional neural network and recurrent neural networks. Similarly, in Xia et al. (2020) defects in Keyhole Tungsten Inert Gas welding were inspected using Resnet (a type of CNN) to recognize different welding states, including burn through, undercut, incomplete penetration and others. Similarly, defect inspection based on deep learning and Hough Transform (HT) was studied in Wang et al. (2019), where researchers used the Gaussian filter for limiting the random noise in obtained images and then used HT for extracting a

Region of Interest clear of useless background in the image. The identification module used a convolutional neural network and the method was reported to be a good balance between the accuracy and computational load.

2.9 Use of Computational Intelligence in Cyber-Physical Systems

The notion of a *Cyber-Physical System (CPS)* is pervasive to every kind of Industry 4.0 concept description. As such, it was introduced some years prior to the introduction of I4.0 term itself (2006 vs. 2011). Both terms are just natural outcomes of the increased extent of digitalization within the industry (and other areas) in general. While original definitions of “systems using computation and communication deeply embedded in and interacting with physical processes to add new capabilities to physical systems” (CPS report 2008; Song et al. 2016) were appropriate, more refined definitions were deemed necessary to better distinguish between CPS and non-CPS systems. An interesting approach to this issue is offered in Song et al. (2016), where four key aspects are taken into account when characterizing CPS:

- **technical emphasis,**
- **cross-cutting aspects,**
- **level of automation,**
- **life-cycle integration.**

Those aspects are not size-related and CPS systems may include miniature systems as well as large-scale and complex ones.

The first of the aforementioned aspects is also one of the most obvious since the term itself implies the interaction of the physical and cybernetic world. It has to be noted that the interaction between physical (in this case mechanical) part of a system and cybernetic (in this case computational) part of a system to enhance its capabilities has been known for a long time in *mechatronic systems*. However, a new dimension was added to this by implicit inclusion of *connectivity* of those systems to

allow for their mutual communication. The sheer extent of connectivity in CPS makes it necessary to consider many aspects from different (even nontechnical) fields, including *security* and *legislation* which are a part of cross-cutting aspects. In addition to that, it is obvious that CPS is designed with a significant degree of automation in its functionality but the human input at a certain level is always expected. This is incorporated into the level of automation aspect of CPS. Since CPS encompasses a very large spectrum of various systems with connectivity capabilities as one of their main properties, they can be characterized also by different levels of integration into the management of products, services, and data (Song et al. 2016; Napoleone et al. 2020).

As pointed out in Panetto et al. (2019) cyber-physical systems in manufacturing (but with some generalization also in other fields) face many challenges under the concept of Industry 4.0 like highly customized supply network control, creation of resilient enterprise to better cope with possible risks, scheduling, and control of digital manufacturing networks or collaborative control. To meet such requirements, it is necessary to apply techniques that allow systems to adapt or learn together with the possibility of self-organization, fault-tolerance as well as handling uncertainty at various levels. With the assumption of vast amounts of data generated by CPS and the necessity to meet previously mentioned requirements, the benefits of the use of computational intelligence for cyber-physical systems are obvious.

This is well summarized in Delicato et al. (2020), where the paradigm of smart cyber-physical systems covering intelligent, self-aware, self-managing, and self-configuring pervasive systems is analyzed. As a part of the cross-cutting aspects of CPS, the security of those systems in view of their connectivity is of crucial importance. This issue is often addressed using a computational intelligence-based approach—researchers in Ding et al. (2018) provides a review of recent advances in security control and attack detection of industrial CPS. In addition to statistics-based machine learning methods, authors present also other methods belonging to the area of computational intelligence (reinforcement learning, neural networks, fuzzy systems).

The problem of scheduling is associated with various aspects of CPS and often needs to be handled with advanced computational techniques

to achieve high performance. With application in wireless sensors, this issue was addressed in Leong et al. (2020), where scheduling of sensor transmissions to estimate the states of multiple remote processes was studied. This was formulated as a Markov decision process and Deep Q-Network was used as a solution. The scheduling problem was also researched in Yi et al. (2020) for tasks in multi-processor distributed systems, but this time authors used an ant colony optimization algorithm to enhance the local search ability and improve the quality of the solution.

More application-oriented research related to the use of computational intelligence in cyber-physical systems was presented in Hou et al. (2020), where the CPS framework is introduced to track truckloads in a highway corridor and to trigger the structural health system for bridges. The linking of bridge response to truck weights is carried out using convolutional neural networks and very good performance is reported. Likewise, the cyber-physical framework is used in Zhang et al. (2019) for structural optimization of complex structures in Real-Time using Hybrid Simulations (RTHS). RTHS is used for evaluation of candidate designs and particle swarm optimization algorithm is used for solving an optimization problem. As noted in Zhang et al. (2020a) current islanded microgrids are turning into CPS, which brings with it various kinds of problems like upload interruption problem. In the work, this is addressed with the use of a secondary control strategy based on improved growing and pruning-radial basis function neural network, leading to improved voltage and frequency stability. In Wang et al. (2018c), researchers used a hybrid fuzzy-PID controller which adapts parameters based on environmental and process variables for controlling the secondary loop of a Lead–Bismuth Eutectic eXperimental Accelerator Driven System (LBE-XADS). This system is viewed as a CPS where physical process variables are monitored and processed intelligently to keep the values of safety parameters in the safety range.

2.10 Case Study: Industrial Parts Recognition by Convolutional Neural Networks for Assisted Assembly

The Industry 4.0 concept defines its supporting technologies for example: digital twin, Radio Frequency IDentification (RFID) technology, virtual and augmented reality, cooperative robotics, big data, deep learning and advanced vision systems. The main idea is the implementation of these technologies for full digitalization in the design of production lines and the necessity of changing and deploying asynchronous assembly lines instead of synchronous. Applications of automatized lines can be found in several areas of the industry: consumer electronics, furniture, clothing, and automotive production. Because of the variation in production, it is almost unnecessary for human stuff to interact with the machines during the assembly process. For the Industry 4.0 concept, cooperative robots with advanced vision systems for knowledge extraction were defined as the main element suitable for cooperation with workers as described, for example, in Liu and Wang (2017). Nowadays, the trend is to have highly variable subassemblies, which must be manually assembled due to unmanageable automation and its implementation. In the case of manual assembly of highly variable parts, it is appropriate to use a Virtual (Augmented) Reality (VR/AR) in combination with image processing to simplify and check the assembly process. For example, an anchoring support system using with AR toolkit is described in Takaseki et al. (2015). There is also possibility to use Computer-Aided Design (CAD) 3D models in the approach from CAD assemblies toward knowledge-based assemblies using an intrinsic knowledge-based assembly model (Vilmart et al. 2018). VR/AR is a direct or indirect view of the physical environment with monitored parts. The field of view for workers can be extended with some additional digital data, mostly as text or image. This additional graphical information must be relevant to the object we are looking at. The visible information can be combined from the vision system or other sources for example integrated industrial sensors, RFID systems, or MEMS units (MicroElectroMechanical Systems).

This case study describes a new approach to parts recognition that are not fixed position (different 2D placement and field of view, large scale range with 3D rotation) by convolutional neural networks. Standard industrial vision systems usually cooperate with conveyor systems and recognized parts are placed on the conveyor belt with a fixed distance from the camera lens and they are digitalized only from one side (usually top). These vision systems can cover some invariance, but in a very limited range (2D rotation with placement and very limited scale). Assisted assembly process based on virtual or augmented reality devices has advanced requirements to recognition robustness. It is necessary to reliably recognize and identify parts from every side with different distances and angles from the camera lens. Convolutional neural networks can help solve this complex task without extra demanding on programming as it was presented in Židek et al. (2019a). There are also two novel neural networks, fire-FRD-CNN (Feature Reuse Detection-Convolutional Neural Network) and mobile-FRD-CNN described in Li et al. (2019). A nice review on recent advances in small object detection based on deep learning can be found in Tong et al. (2020). The most problematic part of the usability of convolutional neural networks is the preparation of the input training image set. This monotonous task can be simplified by the automatized generation of the image set from 3D virtual models which was solved in Židek et al. (2019b). This problem is also described in Socher et al. (2012), Su et al. (2015), Sarkar et al. (2017), and Tian et al. (2018). CNN model trained with general samples can be used after transfer learning also for other recognition tasks. So it is possible to use these pre-trained models for recognition of the industrial part with a significant decrease in training time. For example, an interesting applications for recognition of bearing errors using artificial neural networks are described in Pavlenko et al. (2019a, b). Other applications are in the field of quality prediction of manufacturing processes (Hrehova 2016), validation of serviceability of manufacturing systems (Lazár and Husár 2012), intelligent systems in the railway freight management (Balog et al. 2019) and so on.

The main idea of this case study is a combination of standard machine vision algorithms (thresholding with the Region of Interest) and CNN

algorithms for reliable small part recognition in images with higher resolution (HD or 4K). The pretrained models of CNN networks can be used for industrial parts recognition, as for example:

- Inception V2, 3, 4 with SSD extension,
- MobileNet V2, 3 with SSD extension,
- ResNet-50,
- Xception,
- Inception-ResNet-V2.

The methodology of industrial parts recognition for assisted assembly and its implementation divided into three main steps is explained in the block diagram in Fig. 2.12:

- I. step: generation of training samples from virtual 3D models and implementation of standard machine vision algorithms for identification Region Of Interest (ROI),
- II. step: training (evaluation) of CNN models by virtual samples and testing in embedded systems with Accelerated Processing Units (APU),
- III. step: transfer of trained convolutional neural network models to virtual or augmented devices for assisted assembly tasks.

Fig. 2.12 Main steps of CNN implementation to the assisted assembly process

The main novelty in the field of convolutional networks is the methodology of recognition for industrial parts, preparation of samples from virtual 3D models, and increasing reliability for small parts recognition by identification Regions of Interest in images. This methodology after implementation to embedded devices can be transferred to assisted assembly systems based on VR/AR devices for these three main tasks:

- to train employer for new assembly tasks,
- to help operator marking parts for next assembly step,
- to real-time check of a manual assembly task.

2.10.1 Input Samples Generation from 3D Virtual Models

The assembly's parts can be divided into two basic groups: nonstandard parts (machined parts) and standardized parts (nuts, bolts, washers, etc.). The base part of the assembly is the stepper motor. Next there are two plastic parts with different colors produced by rapid prototyping technology connected to the main part by standardized parts (bolts, nuts, washer, and spring). The standardized parts have small dimensions and are assembled to nonstandard parts. All these parts are usually created or generated in 3D design software and are available before the production starts. 3D models of all assembly parts are available, which can help to train the CNN recognition model faster by the generation of the training set from these virtual 3D models. An example of the generated 2D images dataset from 3D assembly virtual models: plastic parts and the standardized stepper motor are shown in Fig. 2.13.

An automatic generation of samples significantly reduces the preparation time of the training set. The Blender visualization software is used for generation of 2D samples. All parts generated from 3D design software must be converted to a universal 3D format. The most suitable format for the Blender visualization software is OBJ format because it supports transfer of an assigned part color. The generated image variation of movement, scale, and rotation is controlled by Blender API via Python script.

Fig. 2.13 Generated 2D image dataset from the virtual 3D virtual model of the assembly

2.10.2 Identification of a Region of Interest for Recognition of Small Parts

The main limitation for the recognition of small parts by CNN models is low input resolution ($224 \times 224 \times 3$ or $299 \times 299 \times 3$). For larger objects, CNN models work reliably. Small objects lose details during the recognition process because high-resolution images (4K or 8K) are automatically downsampled to the default input layer with low resolution. This is one reason why the recognition of small industrial parts in assembly objects is difficult. The inspiration for solving this problem can be taken from the human brain, which solves the same problem with the recognition of small objects by changing the distance from a recognized object. The next very interesting feature of the human brain is its ability to ignore the areas with plain color in a recognition process and focus mainly on places with some pattern. This problem is solved very simply by changing the position and distance from the recognized object. But this approach is not suitable for industrial tasks.

Two much more effective methods for industry are useful:

1. The vision system with an automated optical zoom (suitable for recognition of parts for long distances).
2. The vision system with a high-resolution camera and integrated identification of the Region of Interest.

The first approach is not suitable for the assisted assembly process because the field of view is usually very near to assembly and the optical zoom procedure is a very time-consuming task. This approach can be used in automatized security camera systems because the detected object can be very far. The second approach based on high-resolution camera, for example, 4K or more, combined with parsing image to set of small regions of interest is much more effective for industrial part recognition. This method has a prerequisite of reliable object detection with minimal delays. Standardized parts with minimal dimensions (e.g., screws, nuts, washers, holes, threads, etc.) used in assembly or before assembly process can be recognized. It is also more easily implemented to virtual devices for assisted assembly tasks.

The process of extraction Region of Interest to identify where industrial parts are located can be realized by these standard machine vision algorithms:

- the Gradient algorithm to isolate clusters of pixels,
- the Contours algorithm to define borders of objects,
- the Closing Square algorithm to increase objects size,
- the Thresholding to reduce noise pixels from image,
- the Region of Interest to localize places of clusters.

An example of the testing input image, its processed image and the final image with thresholding and regions is shown in Fig. 2.14. There are detected six regions where some small parts can be located. This operation reduces image resolution for the CNN model input to 30% and increases input resolution for every feature during the detection process.

Fig. 2.14 Real image of assembly (a), the processed image (b), the final image after thresholding with identification of ROIs (c)

2.10.3 Convolutional Network Transform Learning

The Inception CNN model was selected for the experimental testing. Two separate CNN models (Faster RCNN Inception V2 SSD trained by Common Objects in Context (COCO) dataset) were tested. The first CNN model was used for training on non-standardized parts and the second one on DIN-standardized parts. The timelines of the training process for classification and position losses for both CNN models are shown in Fig. 2.15.

An example of recognition results from testing with virtual photorealistic images and real part images is shown in Fig. 2.16.

The results of the recognition process after CNN transfer learning are shown in Table 2.1.

Training times has been significantly reduced under 2 hours because transfer learning techniques were used. The minimal recognition classification precision decreases after testing with real part images about 30%, which is still acceptable for assisted assembly tasks.

2.10.4 Implementation into Devices for Assisted Assembly

A standard CPU doesn't have enough power to process tasks as image capture, basic filtering, and CNN model execution. So the first step is

Fig. 2.15 Training process for: classification (a)/position (b) loss of unstandardized assembly parts, classification (c)/position (d) loss of standardized assembly parts

Fig. 2.16 Results of experiments with recognition reliability of trained CNN models

Table 2.1 Table with recognition results from all tested CNN models

CNN model used for transfer learning	Training time (TL) hours [h]	Minimal recognition precision virtual parts [%]	Minimal recognition precision real parts [%]
Inception v2 SSD for nonstandard parts	1.3	97	69
Inception v2 SSD for standardized parts	1.5	96	73

testing trained CNN models in embedded devices with support of acceleration neural network execution unit. The next step is visualization in virtual or augmented reality devices.

2.10.4.1 Implementation into Embedded Devices

The first testing platform was an embedded board with integrated APU (GPU with Tensors) Nvidia Xavier development kit with Ubuntu OS Linux distribution as is shown in Fig. 2.17a. The 4K images are acquired by E-Cons dual-camera system with 13Mpix resolution as Continual Service Improvement (CSI) module, mounted in the experimental stand, and rapid prototyped holders. The second testing platform is embedded board Raspberry PI4, which doesn't include any APU unit. Additional

Fig. 2.17 Embedded devices with the implementation of the convolutional neural network (a) Nvidia AGX with E-Cons dual-camera (b) Raspberry PI 4 with CSI camera

computing power for CNN acceleration is acquired by the USB Neural Compute Stick Movidius 2 special module from Intel, which is shown in Fig. 2.17b.

The TensorFlow Framework from Google version 1.15 was used for training all CNN models. Nvidia provides SDK manager with Tensor RT library for trained CNN model to accelerate NVIDIA Xavier embedded device during execution of the CNN model. Intel offers Open VINO toolkit for CNN model acceleration by Intel Movidius USB compute stick combined with Raspberry PI 4. The Open CV library version 4.1 is a universal framework and is used on both platforms for the Region of Interest detection.

2.10.4.2 Implementation to VR/AR Devices

The validated CNN model can be implemented into virtual device HTC Vive Pro for assisted assembly tasks, which provides higher performance for CNN model execution because it uses standard PC with the dedicated graphics card. Standalone augmented devices can be used for simpler assisted assembly tasks, as for example Epson Moverio BT350 with integrated Android board. Both solutions are shown in Fig. 2.18.

The visualization data from the recognition process is realized by the Unity 3D engine, which doesn't have direct support for the CNN model, but it can communicate with the OpenCV framework by the

Fig. 2.18 Assisted assembly (a) Virtual Reality device HTC Vive Pro and Leap Motion, (b) Augmented Reality device Epson Moverio BT350

deep neural network (DNN) library. The Unity 3D creates a PC application for virtual reality device HTC Vive Pro and Android application for augmented reality device Epson Moverio BT350.

To summarize, two CNN models have been designed and tested: the one for nonstandard parts and the second for small standardized parts with single-shot detection algorithm for localization in the plane. The main reason for the preparation of two different CNN models is the reusability CNN model for the standardized part, which can be used for other assemblies. The first convolutional neural network model acquires precision with real parts classification minimum of about 69%. The second CNN model had better accuracy in classification after extraction of the Region of Interest with a minimum of 73%. The future works will be implementation of the Segmentation algorithm included in the TensorFlow version 2, which replaces simple Single-Shot Detection (SSD) algorithms to help detect the exact shape of the object for precise orientation detection in the workspace.

2.11 Discussion

In this chapter, we focused on the use of specific CI paradigms in the context of Industry 4.0. Since both of these areas can be considered very large, we limited to only the most important concepts. It is important to note that the definite consensus of what exactly constitutes each of these fields is lacking. To establish a basic framework for the chapter, we used the major classification presented in Sumathi et al. (2018) and on the lower level we identified six well-established paradigms—neural networks, fuzzy logic, evolutionary computation, swarm intelligence, artificial immune systems, and deep learning—each of which holds significant potential for the design of intelligent systems (Table 2.2). The inclusion of a high number of various novel nature-inspired metaheuristics was avoided since in many cases the benefit of using them compared to the better-established techniques may be questionable. Similarly, the concept of Industry 4.0 encompasses several major technologies and a number of components, where the use of advanced computational techniques is naturally assumed to meet the stringent requirements for high

Table 2.2 Summary of application areas within Industry 4.0 using computational intelligence

CI paradigm	Application area within I4.0	References	CI paradigm	Application area within I4.0	References
Neural Networks	Smart Manufacturing	Mehami et al. (2018) Shi et al. (2019) Wong and Yu (2019)	Swarm Intelligence	CPS	Yi et al. (2020) Zhang et al. (2019)
	CPS	Zhang et al. (2020a)			
	Robotics	Yang et al. (2019)			
	IoT	Djedidi and Djeziri (2020) Veerasamy et al. (2020) Hernández et al. (2020)			
Fuzzy logic	Big Data		Artificial Immune Systems	Smart Manufacturing	Djenouri et al. (2019) Abdi and Feizi-Derakhshi (2020) Nguyen et al. (2020) Wang et al. (2018b) Akram and Raza (2018)
	IoT	Thangaramya et al. (2019) Bu (2018)			

CI paradigm	Application area within I4.0	References	CI paradigm	Application area within I4.0	References
Evolutionary Computation	Big Data	Shukla et al. (2020) Zhang et al. (2020b)		IoT	Semwal and Nair (2020)
	Smart Manufacturing	Huo et al. (2020) Lu and Liu (2018) Chen et al. (2020)		Cybersecurity	Zhang et al. (2011) Aldahheri et al. (2020) Costa Silva et al. (2017)
	CPS	Wang et al. (2018c)	Deep Learning	Fault Detection and Isolation Smart Manufacturing	Wang et al. (2018a) Wang et al. (2020b) Xia et al. (2020) Wang et al. (2019)
	IoT	Tang and Xin (2016)		Cybersecurity	Almiani et al. (2020)
	Smart Manufacturing	Dilberoglu et al. (2017) Mele and Campana (2020) Ewald et al. (2018) Zheng et al. (2020) Nguyen et al. (2019) Saif-Eddine et al. (2019)		CPS	Andersen et al. (2019) Wang et al. (2020a) Leong et al. (2020) Hou et al. (2020) Wang et al. (2020b) Anbarasan et al. (2020) Essiet et al. (2019)
				Big Data	

performance. As shown in Table 2.2, again a smaller number of such application areas was identified so that the use of CI in those works can be easier to generalize. On the other hand, a conceptual similarity of those areas was not taken into account—so some of them may be more general than the others. The most important application areas within Industry 4.0 in terms of their connection with computational intelligence techniques are smart manufacturing, Internet-of-Things, CPS, and Big Data Analytics.

In the case of neural networks, shallow and deep architectures were evaluated separately with the DNNs included in “Deep Learning” Sect. 2.8. Even though the DNNs are currently one of the most promising CI paradigms for many types of problems, “classical” (i.e., shallow) neural networks are still used for various applications. The works summarized in Table 2.2, confirm that the use of network types like RBF or Hopfield still offers attractive properties, e.g., for the regression problems in the context of Industry 4.0. The development of Type-2 fuzzy systems allowed for better handling of uncertainty, for which fuzzy logic is typically applied. Their use, whether in the form of Type-1 or Type-2, ranges from the nonlinear control to machine learning techniques like clustering. These methods are especially suitable for big data analytics, where advanced data mining techniques for finding patterns in vast amounts of data are of crucial importance. As a metaheuristic, evolutionary computation is a fine candidate for the optimization problems with no special requirements for their knowledge. The reviewed works show that this can be used in a wide range of problems, from the supply chain management through the optimization of energy efficiency in wireless networks to additive manufacturing. Swarm intelligence techniques like particle swarm optimization can be actually used in a similar way to EC methods like genetic algorithms or differential evolution. On the other hand, their source of inspiration (bird flocks or ant colonies, etc.) is a tempting solution also for decentralized bio-inspired control of many simple agents (like in networks). Particular tasks within areas like IoT or smart manufacturing in general include workflow scheduling, analysis of complex networks, or even path planning of AGVs in smart factories. Another type of bio-inspired computational paradigm is artificial immune systems, which, in addition to the previously mentioned

optimization or data mining problems, is also used for the area of cybersecurity. This use is quite natural and conforms to the idea of natural immune systems providing a defense against harmful pathogens, and may be of benefit in complex networks (e.g., smart grids and similar).

As mentioned several times in the text, the concept of deep learning, in general, is at this time considered one of the most perspective CI techniques for applications where large amounts of data are present. Therefore, it is closely tied to the very idea of Industry 4.0 and can be expected to become even more powerful with further progress in the hardware. In addition to the use of CNNs, which are the networks typically used in computer vision applications, deep reinforcement learning is particularly interesting, e.g., for robot control in an uncertain environment or handling large sensor networks.

To illustrate the benefit of using deep neural networks in the product manufacturing scenario, a case study of CNNs used for parts recognition in assisted assembly task was introduced. The main advantage of this approach was the possibility of creating training datasets using virtual models. After proper training of the CNN, the solution was implemented in AR/VR devices. The results confirm the viability of the proposed method for the tasks of assisted assembly.

2.12 Conclusion and Future Prospects

This short review attempts at summarizing the use of certain computational intelligence paradigms in the concept of Industry 4.0. However, due to the limited space, only some fundamental paradigms were addressed since the spectrum of bio-inspired computation methods that are applicable within I4.0 is very large. What we tried to address were some of the well-known approaches in CI, which proved to be effective in many different fields and hold significant potential for the use in smart manufacturing. We need to be aware of the fact that the area of computational intelligence is subject to very intense research, making it difficult to capture all its capabilities in a given instant. What remains firmly set and important for the concept of I4.0 is a data-driven aspect of CI methods, which makes it naturally suited for key aspects of I4.0

like cyber-physical systems and big data. Huge amounts of data associated with the use of countless interconnected devices make the methods and models capable of processing it and extracting meaningful information for either finding solutions to the problems or making decisions almost indispensable. In this regard, deep learning is currently one of the most perspective paradigms for many applications in I4.0. Rapid advancements in this particular paradigm have been caused mainly by the availability of very powerful hardware (like GPGPUs—General-Purpose computing on Graphics Processing Units) as well as the aforementioned huge amounts of data. Although there is nothing fundamentally new about the deep neural networks, the lack of powerful enough hardware together with the absence of effective training methods for very large networks made it difficult to obtain good results with them. Fuzzy logic and fuzzy systems have also solidified their position in future applications through the recent developments in Type-2 fuzzy logic, which helps to better tackle uncertainty in data. As such, they can offer a very important advantage over the purely black-box approaches, i.e. the interpretability of the results, which can be of great importance in many fields.

What has not been emphasized in the chapter but is also extremely important with regard to computational intelligence techniques, is their performance boost through the hybridization. Starting from the neuro-fuzzy approaches with which we can obtain interpretable models with NN-like learning and possibly ending with the search of (quasi)optimal parameters or hyperparameters of the models like neural networks, fuzzy systems, support vector machines, and others with bio-inspired non-gradient optimization methods. The methods like particle swarm optimization, genetic algorithm, differential evolution, artificial immune system offer a way to attack many various problems with minimal knowledge. Even though it might be sometimes difficult to explain why they actually work, many NP-hard problems are intractable using conventional computational techniques, making the prospect of having at least some (acceptable) solution attractive. It is of note that many of those methods may serve as an inspiration also on another level—swarm intelligence-based methods are of interest due to the cooperation of simple agents that gives rise to very tempting features like self-organization and self-learning. Such features are certainly more than desirable in the

context of a multitude of embedded devices communicating with each other.

To show a possible application of some of the recent CI techniques, we presented a case study of deep learning paradigm in computer vision. This is one of the most striking examples of successful use of deep neural networks in the area of manufacturing, where the tasks of product inspection for possible defects are of extreme importance. The development of hardware specially designed for handling the tasks of DNN training in these applications allows us to achieve high performance, required for effective use in the industrial area.

A number of the most recent works in I4.0-related research included in this short review attest to the great interest of researchers in CI paradigms. This fact is fully in accordance with the importance of AI within the concept of I4.0—actually AI is so deeply rooted in the basic idea of I4.0 that we can safely say that it is one of its pillars. With this in mind, it is obvious that the actual implementation of I4.0 in current and future factories is also dependent on the success of the implementation of some of the CI paradigms in given applications.

A number of the most recent works in I4.0-related research included in this short review attest to the great interest of researchers in CI paradigms. This fact is fully in accordance with the importance of artificial intelligence within the concept of I4.0—actually AI is so deeply rooted in the basic idea of I4.0 that we can safely say that it is one of its pillars. With this in mind, it is obvious that the actual implementation of I4.0 in current and future factories is also dependent on the success of the implementation of some of the CI paradigms in given applications.

Even though we are still a relatively long way from having the concept of I4.0 implemented in majority of enterprises, even more advanced concepts keep springing up in academic sources. In one of the visions (Demir et al. 2019) for I5.0, very close interaction between humans and robots is assumed. While this is becoming reality also today through a gradual use of collaborative robots, we still cannot talk about the Human-Robot Interaction (HRI) as a natural aspect of the manufacturing process. It is obvious that any advances in the field of HRI are closely bound with the advances in artificial intelligence since in this

interaction we are certainly looking for machines that are safe and close to us in our abilities to adapt and learn.

Each of the commonly accepted three basic pillars of CI (neural networks, fuzzy logic and evolutionary computation) has been subject to intense research in the last decades. Nevertheless, it seems that deep neural networks and DL in general are currently the paradigms that are seen as holding the greatest potential for future applications of intelligent systems. Despite the fact that the best results probably have been achieved in the fields of vision systems as well as voice recognition, possible benefits of DL application can be found in many other fields. On the other hand, possible hybridization of various CI techniques makes it possible to further enhance the performance of intelligent systems in which they are applied. This is especially true for any kind of NP-hard problems found in many applications within I4.0 (or more advanced) concepts, where metaheuristics can be effectively used. In addition to that, DL techniques can be potentially hybridized with fuzzy logic to form so-called deep fuzzy neural networks that fuse the capabilities of neural networks with our way of reasoning. Together with the availability of huge amounts of data, such powerful fusions allow us to take the capabilities of future intelligent machines much closer to humans.

References

- Abdi, Y., and M.-R. Feizi-Derakhshi. 2020. Hybrid multi-objective evolutionary algorithm based on Search Manager framework for big data optimization problems. *Applied Soft Computing* 87 (February 1): 105991.
- Aggarwal, C.C. 2018. Neural networks and deep learning: A textbook [Internet]. NY, USA: Springer International Publishing; 2018 [cited 2020 June 18]. Available from: <https://www.springer.com/gp/book/9783319944623>.
- Agrafiotis, D., and W. Cedeño. 2002. Feature selection for structure—Activity correlation using binary particle swarms. *Journal of Medicinal Chemistry* 45 (March 1): 1098–1107.

- Aickelin, U., and S. Cayzer. 2002. The danger theory and its application to artificial immune systems. *Proceedings of the First International Conference on Artificial Immune Systems*, 141–148.
- Akram, M., and A. Raza. 2018. Data for: Towards the development of robot immune system: A combined approach involving innate immune cells and T-lymphocytes. *BioSystems* [Internet]. Mendeley, August 30 [cited 2020 June 17]; 1. Available from: <https://data.mendeley.com/datasets/5684z3wztx>.
- Alghaheri, S., D. Alghazzawi, L. Cheng, A. Barnawi, and B. Alzahrani. 2020. Artificial immune systems approaches to secure the internet of things: A systematic review of the literature and recommendations for future research. *Journal of Network and Computer Applications* 157 (May 1): 102537.
- Almiani, M., A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque. 2020. Deep recurrent neural network for IoT intrusion detection system. *Simulation Modelling Practice and Theory* 101 (May 1): 102031.
- Anbarasan, M., B. Muthu, C.B. Sivaparthipan, R. Sundarasekar, S. Kadry, S. Krishnamoorthy, et al. 2020. Detection of flood disaster system based on IoT, big data and convolutional deep neural network. *Computer Communications* 150 (January 15): 150–157.
- Andersen, R.E., S. Madsen, A.B.K. Barlo, S.B. Johansen, M. Nør, R.S. Andersen, et al. 2019. Self-learning processes in smart factories: Deep reinforcement learning for process control of robot brine injection. *Procedia Manufacturing* 38 (January 1): 171–177.
- Angeline, P.J. 1998. Using selection to improve particle swarm optimization. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360). Anchorake, USA. pp. 84–89.
- Antão, R. 2017. Type-2 Fuzzy logic: Uncertain systems' modeling and control [Internet]. Springer Singapore [cited 2020 June 18]. Available from: <https://www.springer.com/gp/book/9789811046322>.
- Balog, M., H. Sokhatska, and A. Iakovets. 2019. Intelligent systems in the railway freight management. *Lecture Notes in Mechanical Engineering*, 390–405.
- Bianchi, F.M., E. Maiorino, M.C. Kampffmeyer, A. Rizzi, and R. Jenssen. 2017. *Recurrent neural networks for short-term load forecasting: An overview and comparative analysis* [Internet]. Springer International Publishing [cited 2020 June 18]. Available from: <https://www.springer.com/gp/book/9783319703374>.

- Blum, C., and D. Merkle (eds.). 2008. *Swarm intelligence: Introduction and applications* [Internet]. Berlin Heidelberg: Springer-Verlag [cited 2020 June 15]. Available from: <https://www.springer.com/gp/book/9783540740889>.
- Bonabeau, E., G. Theraulaz, and M. Dorigo. 1999. *Swarm intelligence: From natural to artificial systems*, 1st ed. New York, NY: Oxford University Press.
- Bu, F. 2018. An efficient fuzzy c-means approach based on canonical polyadic decomposition for clustering big data in IoT. *Future Generation Computer Systems* 88 (November 1): 675–682.
- Cantú-Paz, E. 1998. A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis* 10 (2): 141–171.
- Castro, L.N., and J. Timmis. 2002. An artificial immune network for multimodal function optimization. Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), pp. 699–704, Vol. 1.
- Castro, L.N., and F.J.V. Zuben. 2000. The clonal selection algorithm with engineering applications. Workshop Proceedings of GECCO, 7.
- Chang, C.S., and D.Y. Xu. 2000. Differential evolution based tuning of fuzzy automatic train operation for mass rapid transit system. *IEE Proceedings—Electric Power Applications* 147 (3, May): 206–212.
- Chen, Z., X. Ming, T. Zhou, and Y. Chang. 2020. Sustainable supplier selection for smart supply chain considering internal and external uncertainty: An integrated rough-fuzzy approach. *Applied Soft Computing* 87 (February 1): 106004.
- Chiou, J.-P., and F.-S. Wang. 1998. A hybrid method of differential evolution with application to optimal control problems of a bioprocess system. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), pp. 627–632.
- Costa Silva, G., W.M. Caminhas, and R.M. Palhares. 2017. Artificial immune systems applied to fault detection and isolation: A brief review of immune response-based approaches and a case study. *Applied Soft Computing* 57 (August 1): 118–131.
- CPS report. 2008. CPS-Summit. Holistic approaches to cyber-physical integration [Internet]. CPSWeek report [cited 2020 June 18]. Available from: https://iccps2012.cse.wustl.edu/_doc/CPS_Summit_Report.pdf.
- de Castro, L.N. 2006. *Fundamentals of natural computing: Basic concepts, algorithms, and applications*, 1st ed. Boca Raton: Chapman and Hall/CRC.
- de Castro, L.N., and F.J.V. Zuben. 2002. *aiNet: An artificial immune network for data analysis* [Internet]. Data mining: A heuristic approach. IGI Global

- [cited 2020 June 15]. pp. 231–260. Available from: www.igi-global.com/chapter/data-mining-heuristic-approach/7592.
- Delicato, F.C., A. Al-Anbuky, and K.I.-K. Wang. 2020. Editorial: Smart cyber-physical systems: Toward pervasive intelligence systems. *Future Generation Computer Systems* 107 (June 1): 1134–1139.
- Demir, K.A., G. Döven, and B. Sezen. 2019. Industry 5.0 and human-robot co-working. *Procedia Computer Science* 158 (January 1): 688–695.
- Dewang, H.S., P.K. Mohanty, and S. Kundu. 2018. A robust path planning for mobile robot using smart particle swarm optimization. *Procedia Computer Science* 133 (January 1): 290–297.
- Dilberoglu, U.M., B. Gharehpapagh, U. Yaman, and M. Dolen. The role of additive manufacturing in the era of Industry 4.0. *Procedia Manufacturing* 11 (January 1): 545–554.
- Ding, D., Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang. 2018. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing* 275 (January 31): 1674–1683.
- Djedidi, O., and M.A. Djeziri. 2020. Power profiling and monitoring in embedded systems: A comparative study and a novel methodology based on NARX neural networks. *Journal of Systems Architecture* 111 (December 1): 101805.
- Djenouri, Y., D. Djenouri, A. Belhadi, P. Fournier-Viger, J. Chun-Wei Lin, and A. Bendjoudi. 2019. Exploiting GPU parallelism in improving bees swarm optimization for mining big transactional databases. *Information Sciences* 496 (September 1): 326–342.
- Dorigo, M., and G. Di Caro. 1999. Ant colony optimization: A new meta-heuristic. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), pp. 1470–1477, Vol. 2.
- Engelbrecht, A.P. 2007. *Computational intelligence: An introduction*, 2nd ed. West Sussex: Wiley.
- Essiet, I., Y. Sun, and Z. Wang. 2019. Big data analysis for gas sensor using convolutional neural network and ensemble of evolutionary algorithms. *Procedia Manufacturing* 35 (January 1): 629–634.
- Ewald, A., T. Sassenberg, and J. Schlattmann. 2018. Evolutionary-based optimization strategy in a hybrid manufactured process using LMD. *Procedia CIRP* 74 (January 1): 163–167.
- Fang, W., J. Sun, Y. Ding, X. Wu, and W. Xu. 2010. A review of quantum-behaved particle swarm optimization. *IETE Technical Review* 27 (4, July 1): 336–348. Taylor & Francis.

- Feoktistov, V. 2006. *Differential evolution: In search of solutions* [Internet]. Springer US [cited 2020 June 15]. Available from: <https://www.springer.com/gp/book/9780387368955>.
- Fonlupt, C., J.-K. Hao, E. Lutton, E. Ronald, and M. Schoenauer. 2006. Artificial evolution: 4th European conference. AE'99 Dunkerque, France, November 3–5, 1999 Selected Papers. Springer.
- Fulcher, J. (ed.). 2008. *Computational intelligence: A compendium* [Internet]. Berlin Heidelberg: Springer-Verlag [cited 2020 June 15]. Available from: <https://www.springer.com/gp/book/9783540782926>.
- Gill, S.S., R. Buyya, I. Chana, M. Singh, and A. Abraham. 2018. BULLET: Particle swarm optimization based scheduling technique for provisioned cloud resources. *Journal of Network and Systems Management* 26 (2, April 1): 361–400.
- Goldberg, D., K. Deb, H. Kargupta, and G. Harik. 1995. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. *Morgan Kaufmann*, pp. 56–64.
- Goldberg, D.E., and L. Wang. Adaptive niching via coevolutionary sharing. In *Genetic algorithms and evolution strategy in engineering and computer science*, Chapter 2, 21–38. West Sussex: Wiley.
- Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep learning*. MIT Press.
- Haykin, S.S. 2009. *Neural networks and learning machines*. New York: Prentice Hall.
- Hendtlass, T. 2001. A combined swarm differential evolution algorithm for optimization problems. In *Engineering of intelligent systems*, ed. L. Monostori, J. Váncza, and M. Ali, 11–18. Berlin, Heidelberg: Springer.
- Hernández, G., E. Zamora, H. Sossa, G. Téllez, and F. Furlán. 2020. Hybrid neural networks for big data classification. *Neurocomputing* 390 (May 21): 327–340.
- Hong, Y., L. Chen, and L. Mo. 2019. Optimization of cluster resource indexing of Internet of Things based on improved ant colony algorithm. *Cluster Computing* 22 (3, May 1): 7379–7387.
- Hou, R., S. Jeong, J.P. Lynch, and K.H. Law. 2020. Cyber-physical system architecture for automating the mapping of truck loads to bridge behavior using computer vision in connected highway corridors. *Transportation Research Part C: Emerging Technologies* 111 (February 1): 547–571.
- Hrehova, S. 2016. Predictive model to evaluation quality of the manufacturing process using Matlab tools. *Procedia Engineering*, 149–154. Elsevier Ltd.

- Hrstka, O., and A. Kučerová. 2004. Improvements of real coded genetic algorithms based on differential operators preventing premature convergence. *Advances in Engineering Software* 35 (3, March 1): 237–246.
- Huo, J., F.T.S. Chan, C.K.M. Lee, J.O. Strandhagen, and B. Niu. 2020. Smart control of the assembly process with a fuzzy control system in the context of Industry 4.0. *Advanced Engineering Informatics* 43 (January 1): 101031.
- Ippolito, M.G., E.R. Sanseverino, and F. Vuinovich. 2004. Multiobjective ant colony search algorithm optimal electrical distribution system planning. Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753). Portland, USA, IEEE, pp. 1924–1931, Vol. 2.
- Iqbal, R., F. Doctor, B. More, S. Mahmud, and U. Yousuf. 2020a. Big data analytics and computational intelligence for cyber-physical systems: Recent trends and state of the art applications. *Future Generation Computer Systems* 105 (April 1): 766–778.
- Iqbal, R., F. Doctor, B. More, S. Mahmud, and U. Yousuf. 2020b. Big data analytics: Computational intelligence techniques and application areas. *Technological Forecasting and Social Change* 153 (April 1): 119253.
- Karray, F.O., and C.W.D. Silva. 2004. *Soft computing and intelligent systems design: Theory, tools and applications*, 1st ed. Harlow, UK and New York: Addison-Wesley.
- Kennedy, J., and R. Eberhart. 1995. Particle swarm optimization. Proceedings of ICNN'95—International Conference on Neural Networks. Perth, Australia: IEEE, pp. 1942–1948, Vol. 4.
- Khan, M., B. Jan, and H. Farman. 2019. Deep learning: Convergence to Big Data analytics [Internet]. Springer Singapore [cited 2020 June 17]. Available from: <https://www.springer.com/gp/book/9789811334580>.
- Kim, J., and P.J. Bentley. 2002. Towards an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection. Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), pp. 1015–1020, Vol. 2.
- Kim, P. 2017. MATLAB deep learning. Seoul, Korea: Apress.
- Knight, T., and J. Timmis. 2002. A multi-layered immune inspired approach to data mining. Proceedings of the 4th International Conference Recent Advances in Soft Computing pp. 266–271.
- Lazár, Ivan, and J. Husár. 2012. Validation of the serviceability of the manufacturing system using simulation. *Journal on Efficiency and Responsibility in Education and Science* 5 (4): 252–261. Czech University of Life Sciences Prague.

- Lee, J., H. Davari, J. Singh, and V. Pandhare. 2018. Industrial artificial intelligence for Industry 4.0-based manufacturing systems. *Manufacturing Letters* 18 (October 1): 20–23.
- Leong, A.S., A. Ramaswamy, D.E. Quevedo, H. Karl, and L. Shi. 2020. Deep reinforcement learning for wireless sensor scheduling in cyber–physical systems. *Automatica* 113 (March 1): 108759.
- Li, W., K. Liu, L. Yan, F. Cheng, Y.Q. Lv, and L.Z. Zhang. 2019. FRD-CNN: Object detection based on small-scale convolutional neural networks and feature reuse. *Scientific Reports* 9 (1, December 1). Nature Publishing Group.
- Liu, H., and L. Wang. 2017. An AR-based worker support system for human-robot collaboration. *Procedia Manufacturing* 11: 22–30.
- Liu, J. 2013. Radial Basis Function (RBF) neural network control for mechanical systems: Design, analysis and Matlab simulation [Internet]. Berlin Heidelberg: Springer-Verlag [cited 2020 June 24]. Available from: <https://www.springer.com/gp/book/9783642348150>.
- Lones, M. 2014. *Metaheuristics in nature-inspired algorithms*, 4. Vancouver: Canada.
- Lu, T., and S.-T. Liu. 2018. Fuzzy nonlinear programming approach to the evaluation of manufacturing processes. *Engineering Applications of Artificial Intelligence* 72 (June 1): 183–189.
- Mariano, C.E., and E. Morales. 1999. A multiple objective Ant-Q algorithm for the design of water distribution irrigation networks. *Technical Report HC-9904*, 41.
- Mehami, J., M. Nawi, and R.Y. Zhong. 2018. Smart automated guided vehicles for manufacturing in the context of Industry 4.0. *Procedia Manufacturing* 26 (January 1): 1077–1086.
- Mele, M., and G. Campana. 2020. Sustainability-driven multi-objective evolutionary orienting in additive manufacturing. *Sustainable Production and Consumption* 23 (July 1): 138–147.
- Messerschmidt, L., and A.P. Engelbrecht. 2004. Learning to play games using a PSO-based competitive learning approach. *IEEE Transactions on Evolutionary Computation* 8 (3, June): 280–288.
- Miller, B.L., and M.J. Shaw. 1996. Genetic algorithms with dynamic niche sharing for multimodal function optimization. *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 786–791.
- Munakata, T. 2008. *Fundamentals of the new artificial intelligence: Neural, evolutionary, fuzzy and more* [Internet], 2nd ed. London: Springer-Verlag [cited 2020 June 24]. Available from: <https://www.springer.com/gp/book/9781846288388>.

- Napoleone, A., M. Macchi, and A. Pozzetti. 2020. A review on the characteristics of cyber-physical systems for the future smart factories. *Journal of Manufacturing Systems* 54 (January 1): 305–335.
- Nasraoui, O., D. Dasgupta, and F. González. 2002. The promise and challenges of artificial immune system based web usage mining. Proceedings of the Second SIAM International Conference on Data Mining, August 11, pp. 29–32.
- Nayyar, A., D.-N. Le, and N.G. Nguyen. 2018. Advances in Swarm intelligence for optimizing problems in computer science, 1st ed. Boca Raton, FL: Chapman & Hall/CRC.
- Nguyen, B.H., B. Xue, and M. Zhang. 2020. A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation* 54 (May 1): 100663.
- Nguyen, S., D. Thiruvady, A.T. Ernst, and D. Alahakoon. 2019. A hybrid differential evolution algorithm with column generation for resource constrained job scheduling. *Computers & Operations Research* 109 (September 1): 273–287.
- Pampara, G., A.P. Engelbrecht, and N. Franken. 2006. Binary differential evolution. 2006 IEEE International Conference on Evolutionary Computation, pp. 1873–1879.
- Panetto, H., B. Iung, D. Ivanov, G. Weichhart, and X. Wang. 2019. Challenges for the cyber-physical manufacturing enterprises of the future. *Annual Reviews in Control* 47 (January 1): 200–213.
- Pavlenko, I., V. Simonovskiy, V. Ivanov, J. Zajac, and J. Pitel. 2019a. Application of artificial neural network for identification of bearing stiffness characteristics in rotor dynamics analysis. Advances in Design, Simulation and Manufacturing, pp. 325–335.
- Pavlenko, I., J. Trojanowska, V. Ivanov, and O. Liaposhchenko. 2019b. Scientific and methodological approach for the identification of mathematical models of mechanical systems by using artificial neural networks. Innovation, Engineering and Entrepreneurship, pp. 299–306.
- Pedrycz, W., and F. Gomide. 2007. *Fuzzy systems engineering: Toward human-centric computing*, 1st ed. Hoboken, NJ: Wiley-IEEE Press.
- Saif-Eddine, A.S., M.M. El-Beheiry, and A.K. El-Kharbotly. 2019. An improved genetic algorithm for optimizing total supply chain cost in inventory location routing problem. *Ain Shams Engineering Journal* 10 (1, March 1): 63–76.
- Sarkar, K., K. Varanasi, and D. Stricker. 2017. Trained 3D models for CNN based object recognition. Proceedings of the 12th International

- Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Visigrapp 2017), Vol. 5. German Research Center Artificial Intelligence DFKI, Kaiserslautern, Germany. German Res C, pp. 130–137.
- Semwal, T., and S.B. Nair. 2020. A decentralized artificial immune system for solution selection in cyber-physical systems. *Applied Soft Computing* 86 (January 1): 105920.
- Sengupta, S., S. Basak, P. Saikia, S. Paul, V. Tsilavoutis, F. Atiah, et al. 2020. A review of deep learning with special emphasis on architectures, applications and recent trends. *Knowledge-Based Systems* 194 (April 22): 105596.
- Shi, C., J. Qian, W. Zhu, H. Liu, S. Han, and X. Yang. 2019. Nondestructive determination of freshness indicators for tilapia fillets stored at various temperatures by hyperspectral imaging coupled with RBF neural networks. *Food Chemistry* 275 (March 1): 497–503.
- Shukla, A.K., M. Yadav, S. Kumar, and P.K. Muhuri. 2020. Veracity handling and instance reduction in big data using interval type-2 fuzzy sets. *Engineering Applications of Artificial Intelligence* 88 (February 1): 103315.
- Socher, R., B. Huval, B. Bhat, C.D. Manning, and A.Y. Ng. 2012. Convolutional-recursive deep learning for 3D object classification. NIPS.
- Song, H., D.B. Rawat, S. Jeschke, and C. Brecher. 2016. *Cyber-physical systems: Foundations, principles and applications*. Burlington, MA: Morgan Kaufmann.
- Srinivas, N., and K. Deb. 1991. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2 (3): 221–248.
- Stützle, T., and H.H. Hoos. 2000. MAX–MIN ant system. *Future Generation Computer Systems* 16 (8, June 1): 889–914.
- Su, H., C.R. Qi, Y. Li, and L.J. Guibas. 2015. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. Proceedings of the IEEE International Conference on Computer Vision.
- Sumathi, S., T. Hamsapriya, and P. Surekha. 2008. *Evolutionary intelligence: An introduction to theory and applications with Matlab*, 1st ed. Berlin: Springer.
- Sumathi, S., L.A. Kumar, and P. Surekha. 2018. Computational intelligence paradigms for optimization problems using MATLAB®/SIMULINK®, 1st ed. CRC Press.
- Sun, H., W. Jie, J. Loo, L. Wang, S. Ma, G. Han, et al. 2018. A parallel self-organizing overlapping community detection algorithm based on swarm intelligence for large scale complex networks. *Future Generation Computer Systems* 89 (December 1): 265–285.

- Takaseki, R., R. Nagashima, H. Kashima, and T. Okazaki. 2015. Development of anchoring support system using with AR Toolkit. 2015 7th International Conference on Emerging Trends in Engineering & Technology (ICETET). IEEE, pp. 123–127.
- Tan, Y., Y. Shi, and K.C. Tan. 2010. Advances in swarm intelligence: First international conference, ICSI 2010, Beijing, China, June 12–15. Proceedings, Part I [Internet]. Berlin Heidelberg: Springer-Verlag [cited 2020 June 15]. Available from: <https://www.springer.com/gp/book/9783642134944>.
- Tang, M., and Y. Xin. 2016. Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization. *Computer Networks* 100 (May 8): 1–11.
- Thangaramya, K., K. Kulothungan, R. Logambigai, M. Selvi, S. Ganapathy, and A. Kannan. 2019. Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in IoT. *Computer Networks* 151 (March 14): 211–223.
- Thapar, P., and U. Batra. 2018. Implementation of ant colony optimization in routing protocol for internet of things. In Innovations in computational intelligence : Best selected papers of the third international conference on REDSET 2016 [Internet], ed. B. Panda, S. Sharma, and U. Batra, 151–164. Singapore: Springer [cited 2020 June 15]. Available from: https://doi.org/10.1007/978-981-10-4555-4_10.
- Tian, Y., X. Li, K. Wang, and F.Y. Wang. 2018. Training and testing object detectors with virtual images. *IEEE/CAA Journal of Automatica Sinica* 5 (2): 539–546.
- Timmis, J., and M. Neal. 2001. A resource limited artificial immune system for data analysis. In *Research and development in intelligent systems XVII*, ed. M. Bramer, A. Preece, and F. Coenen, 19–32. London: Springer.
- Tong, K., Y. Wu, and F. Zhou. 2020. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, p. 103910. Elsevier Ltd.
- Veerasamy, V., N.I.A. Wahab, R. Ramachandran, B. Madasamy, M. Mansoor, M.L. Othman, et al. 2020. A novel RK4-Hopfield Neural network for power flow analysis of power system. *Applied Soft Computing* 93 (August 1): 106346.
- Vilmart, H., J.-C. Léon, and F. Ulliana. 2018. From CAD assemblies toward knowledge-based assemblies using an intrinsic knowledge-based assembly model. *Computer-Aided Design and Applications* 15 (3, May 4): 300–317.

- Wang, J., P. Fu, and R.X. Gao. 2019. Machine vision intelligence for product defect inspection based on deep learning and Hough transform. *Journal of Manufacturing Systems* 51 (April 1): 52–60.
- Wang, J., Y. Ma, L. Zhang, R.X. Gao, and D. Wu. 2018a. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems* 48 (July 1): 144–156.
- Wang, S., Y.C. Liang, W.D. Li, and X.T. Cai. 2018b. Big Data enabled Intelligent Immune System for energy efficient manufacturing management. *Journal of Cleaner Production* 195 (September 10): 507–520.
- Wang, W., F. Di Maio, and E. Zio. 2018c. Hybrid fuzzy-PID control of a nuclear Cyber-Physical System working under varying environmental conditions. *Nuclear Engineering and Design* 331 (May 1): 54–67.
- Wang, D., H. Deng, and Z. Pan. 2020a. MRCDRL: Multi-robot coordination with deep reinforcement learning. *Neurocomputing* 406 (September 17): 68–76.
- Wang, Q., W. Jiao, P. Wang, and Y. Zhang. 2020b. A tutorial on deep learning-based data analytics in manufacturing through a welding case study. *Journal of Manufacturing Processes* [Internet]. May 27 [cited 2020 June 17]. Available from: <https://www.sciencedirect.com/science/article/pii/S1526612520302668>.
- Wong, S.F., and Z. Yu. 2019. The mobile robot anti-disturbance vSLAM navigation algorithm based on RBF neural network. *Procedia Manufacturing* 38 (January 1): 400–407.
- Xia, C., Z. Pan, Z. Fei, S. Zhang, and H. Li. 2020. Vision based defects detection for Keyhole TIG welding using deep learning with visual explanation. *Journal of Manufacturing Processes* 56 (August 1): 845–855.
- Xie, Y., Y. Zhu, Y. Wang, Y. Cheng, R. Xu, A.S. Sani, et al. 2019. A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment. *Future Generation Computer Systems* 97 (August 1): 361–378.
- Xing, B., and W.-J. Gao. 2014. Innovative computational intelligence: A rough guide to 134 Clever algorithms [Internet]. Springer International Publishing [cited 2020 June 24]. Available from: <https://www.springer.com/gp/book/9783319034034>.
- Yang, M., Z. Du, L. Sun, and W. Dong. 2019. Optimal design, modeling and control of a long stroke 3-PRR compliant parallel manipulator with variable thickness flexure pivots. *Robotics and Computer-Integrated Manufacturing* 60 (December 1): 23–33.

- Yang, X.-S., Z. Cui, R. Xiao, A.H. Gandomi, and M. Karamanoglu (eds.). 2013. *Swarm intelligence and bio-inspired computation: Theory and applications*, 1st ed. Amsterdam and Boston: Elsevier.
- Yi, N., J. Xu, L. Yan, and L. Huang. 2020. Task optimization and scheduling of distributed cyber–physical system based on improved ant colony algorithm. *Future Generation Computer Systems* 109 (August 1): 134–148.
- Zhang, R., B.M. Phillips, P.L. Fernández-Cabán, and F.J. Masters. 2019. Cyber-physical structural optimization using real-time hybrid simulation. *Engineering Structures* 195 (September 15): 113–124.
- Zhang, B., C. Dou, T. Zhang, and Z. Zhang. 2020a. An IGAP-RBFNN-based secondary control strategy for islanded microgrid-cyber physical system considering data uploading interruption problem. *Neurocomputing* 397 (July 15): 422–437.
- Zhang, X., X. Ming, and D. Yin. 2020b. Application of industrial big data for smart manufacturing in product service system based on system engineering using fuzzy DEMATEL. *Journal of Cleaner Production* 265 (August 20): 121863.
- Zhang, Y., L. Wang, W. Sun, R.C. Green II., and M. Alam. 2011. Distributed intrusion detection system in a multi-layer network architecture of smart grids. *IEEE Transactions on Smart Grid* 2 (4, December): 796–808.
- Zheng, J., L. Wang, and J. Wang. 2020. A cooperative coevolution algorithm for multi-objective fuzzy distributed hybrid flow shop. *Knowledge-Based Systems* 195 (April 22): 105536.
- Židek, K., A. Hosovsky, J. Pitel', and S. Bednár. 2019a. Recognition of assembly parts by convolutional neural networks. *Lecture Notes in Mechanical Engineering*.
- Židek, K., P. Lazorík, J. Pitel', and A. Hošovský. 2019b. An automated training of deep learning networks by 3D virtual models for object recognition. *Symmetry*. MDPI AG 11 (4, April 1).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.