Contrasting Human and Computational Intelligence Based Autonomous Behaviors in a Blue–Red Simulation Environment

Shir Li Wang, Kamran Shafi, *Member, IEEE*, Theam Foo Ng, *Member, IEEE*, Chris Lokan, and Hussein A. Abbass, *Senior Member, IEEE*

***Abstract*—Autonomous systems are making their way to the market. The transition from tasks performed by humans to tasks performed by machines begs for an answer to one of the most challenging questions in this area of research: Will humans understand and trust what a machine does? Analyzing human and machine behaviors offers the foundational steps toward finding answers to this question. This paper contributes a novel methodology for transforming low-level actions by each agent into high-level categorization of strategies to contrast the behaviors of humans and machines using a computational red teaming environment with a red (evader) and a blue (pursuer) agent. Two orthogonal sources of uncertainty were examined: the uncertainty in the blue agent's situation awareness about the red, and the red agent's uncertainty resulting from deceptive actions by the blue. For each uncertainty source, two different experiments were conducted by varying the controller of the red agent. In one experiment, the red agent was controlled by one of the 34 human subjects; and in the second, by an evolved neural network. The blue agent was controlled by a scripted rule-based system. In this time-critical task, the results revealed that humans tend to follow systemic and consistent strategies, sometimes ignoring the information available to them. On the other hand, machines tend to evolve more complex and diverse strategies. This finding calls for new computational intelligence techniques to enable the fusion of these different strategies into forms that each party can understand and use effectively.**

***Index Terms*—Computational red teaming, human-machine behavioral analysis transparent artificial intelligence transparent autonomy, neuro-evolution.**

I. Introduction

AUTONOMOUS systems are leaving the laboratory environment, be it in the form of a physical robot, such as a Google car, or in the form of a software robot, such as an internet trader. To establish trust between humans and artificial intelligence (AI) agents, it is pertinent to design computational intelligence techniques that can analyse the behaviour of each entity to identify their similarities and differences (i.e. transparency of the AI model). These computational intelligence techniques have multiple uses, ranging from communicating these differences to humans to allow humans a level of understanding of their own decisions and the AIs' decisions, through to forming a shared and transparent understanding to improve the collaborative environment that enables a symbiotic relationship between humans and AIs.

Characterising human and machine behaviours has significant implications for many disciplines, including decision analysis, AI and human-machine systems. For example, one of the goals in behavioural AI studies is to develop machines and agent-based systems that can produce human-like behaviours at an individual or a collective level in a society [1], [2]. These machines can then be used for improving humans' cognitive competencies through decision aiding and training tools. The effectiveness of these artificial entities could be judged at a much higher level of abstraction by comparing the macro outcomes (i.e., task performance) of human-based operations with those of machine-based operations.

Cooperation in human machine interaction [3] has led to the introduction of the field of human machine cooperation (HMC). In HMC, a machine is not treated as a tool, but as an autonomous AI that works by itself or together with a human partner to act on dynamic situations. Computational intelligence techniques can assist in designing methodologies to analyse the behaviours of humans and machines in these environments.

Agents' interactions sometimes also involve a level of competition, denoting a degree of conflict in the objectives of the agents. Red teaming (RT) is an intuitive approach to studying interactions among a group of entities with conflicting goals in competitive environments. The minimum ingredients of an RT environment are two entities with conflicting objective(s): $blue$ and red. $Blue$ refers to a friendly entity, while red refers to an entity that has the potential to influence $blue$ to hinder it from achieving its own objectives. RT can be traced back to Sun Tzu, where 'playing devil's advocate' was used as a methodology to evaluate one's own decisions and plans [4]–[6]. RT is traditionally used by the military to role-play the enemy, understand

Manuscript received August 19, 2016; revised November 14, 2016 and December 6, 2016; accepted December 14, 2016. Date of publication January 17, 2017; date of current version February 16, 2017.

S. L. Wang is with the Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, Tanjong Malim 35900, Malaysia (e-mail: shirli_wang@yahoo.com).

K. Shafi, C. Lokan, and H. A. Abbass are with the School of Engineering and Information Technology, UNSW-Canberra, Canberra, ACT 2612, Australia (e-mail: k.shafi@adfa.edu.au; c.lokan@adfa.edu.au; h.abbass@adfa.edu.au).

T. F. Ng is with the Centre for Global Sustainability Studies, Universiti Sains Malaysia, Penang 11800, Malaysia (e-mail: tfng@usm.my).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TETCI.2016.2641929

enemy's behaviour, test and evaluate its courses of actions or judgements, assess the vulnerabilities of, and understand the dynamics between, the $blue$ and red teams [4]–[6].

Computational red teaming (CRT) [7], [8] is the term coined when the overall RT activity is carried out in silico, or when a human-based RT exercise is augmented with computational methods and models. A red agent in CRT can be a set of computational models for adversarial behaviours, while the $blue$ agent can be a set of computational models representing the system that needs to be evaluated or protected. This representation of CRT allows us to study the interaction's interface between red and $blue$.

In this study, we used a CRT environment to understand the possible attributes of benchmarking human behaviour against machine behaviour in a simplified CRT environment. It was important to work with a simple environment to design sound experiments to isolate cause and effect. By contrasting the two behaviours, we hoped to identify strategies to improve human decision making. In addition, if machine behaviour is drastically different from humans, understanding the similarities and differences can offer insight into how we could possibly integrate the two. The decision to let an AI act automatically or to combine the actions from humans and AIs is known as function allocation in HMC. Function allocation is crucial in the effectiveness of HMC [3]. Understanding the behaviours of humans and AIs helps improve function allocation.

The contributions of this paper are twofold. First, we present a fine-grained methodology for comparing human and machine behaviours in a CRT time-critical decision-making environment. This methodology offers systematic steps that can enable analysts to compare predator-prey systems [9] in nature as well as human decision making in uncertain environments, information warfare, and cyber security applications [10], [11].

Second, we extend our previous analysis which was limited to the computational software environment [12] to an analysis that contrasts human and AIs' behaviours in the environment. Each agent generates a sequence of actions (i.e., simple movements in a bounded two-dimensional (2D) continuous space) according to different strategies. The $blue$ agent follows scripted strategies in response to red's actions. The red agent, which can be either a human subject ($human\text{-}based\ red$) or a neuro–evolutionary learning controller ($machine\text{-}based\ red$), is free to choose (in the human case) or evolve (in the machine case) its own strategies. We then analyse these strategies by characterising the trajectories taken by the agents, using a measurement methodology based on derivatives, where a derivative refers to the rate of change in an agent's movements.

In [12], we designed, analysed and tested the software environment (machine agent) that we have used in the current study. We have extended this work through:

1) an experiment to compare machines and humans, the design of which is reported and discussed in this paper, including the experimental protocol for collecting human data; and
2) an in-depth analysis that contrasts the behaviours of the humans and machine agents.

We used a $blue\text{-}red$ CRT environment, where humans play red in different configurations defined by information quality and deceptive actions. Our primary hypothesis was that humans would adopt different strategies given different configurations. Our secondary hypothesis was that the machine agent would express different behaviours compared with those displayed by humans. These two hypotheses can be explained more precisely as follows:

1) Noise in sensory inputs, and intentional deviation in the output (or deception) affect the behaviours of human and machine agents.
2) The perception of extra task-relevant stimuli affects an entity's behaviour.
3) Humans use different strategies in different scenarios compared with neuro–evolutionary-based agents.

In summary, this work presents novel methodologies for transforming the low-level actions of each agent into the high-level categorisation of strategies to allow a comparison of strategies exhibited by each agent. Second, the paper's analysis contrasts human and machines behaviours distilling salient features that distinguish both types of behaviours.

The remainder of this paper is organised as follows. Section II describes background information and work related to the concepts presented in this paper, including the CRT. Section III presents our methodology. Results and discussions follow in Section IV. Finally, conclusions are drawn in Section V.

II. Related Work

Research in behavioural decision theory [13] suggests that agents' actions reflect their behaviour. In other words, the actions that are expressed or produced by an agent should reflect the underlying strategy used by the agent for action production. We take this view to analyse an agent's behaviour. Areas such as plan recognition [14], [15], deception detection [16], intent inference [17], and course of action analysis [18], [19] can all offer CRT tools to model and analyse behavioural spaces.

The study of behaviour is not possible without having a context to provide the meaning, the frame of reference, and the knowledge required to carry out a task or activity [20]. When the entities embedded in the activity are humans, prediction of behaviour becomes even more difficult because of individual differences related to perception and processing capacities, prior knowledge, skills, expertise and personality traits. Thus, behaviour is a function of the task, context and individual differences [20]. Abbass [7] factored into the behaviour function the environment in which the task activity is performed; thus situating the analysis within a context and a frame of reference to explain the dynamics.

The context used in the study for this present paper was CRT, which can be used as a risk assessment methodology by analysts and decision makers to explore a space of possibilities, choices and solutions [21], [22]. As mentioned, CRT involves computational methodologies or models to implement or support an RT exercise.

An example of CRT can be found in [21], in which the Multiobjective Evolutionary Based Risk Assessment (MEBRA)

framework was introduced to explore and evaluate different algorithms under risk. MEBRA explores scenarios in which the tested algorithms may perform poorly, known as the failure regime of the algorithms. The use of MEBRA in air traffic management can be found in [23], where MEBRA was used to search scenarios that could pose risks to two algorithms in aircraft landing sequencing. In that research, the tested algorithms were viewed as $blue$ while the scenarios that caused the failures of the algorithm were viewed as red.

CRT builds on a long history of prior literature in different domains and links to other areas of research. Search techniques, such as genetic algorithms (GAs), can offer CRT ways to optimise performance. For example, in an approach named 'adaptive manoeuvring logic', evolutionary algorithms were applied to air combat to learn the behaviour of pilots, and the learned behaviours were used to train pilots, as early as in the 1970s [24]. GAs have also been used to assist military planners in dynamic and changing environments [25], [26]. In recent years, more advanced evolutionary algorithms, in particular co-evolutionary algorithms, have been used to evolve strong strategies at the interaction and interface of the blue and red teams [27]. As such, CRT extends the study of computational models, human factor studies and other disciplines to bring together a methodology for risk assessment, challenging humans or machines in a task and offering an environment to study human-machine interaction in competitive settings.

A further area of research relevant to CRT is adversarial modelling and learning. Many studies in cyber security have focused on the simulation of adversarial attacks faced by a machine learning system, the development of defence strategies against adversarial attacks, or both [28]–[32]. Adversarial learning can be formulated as a two-sided game between learners and adversaries [11], [28], [33]. For example, some machine learning studies [28], [33] formulated the classification problem as a game between classifiers and adversaries, or in the language of CRT, $blue - red$ interaction.

Unfortunately, strategies employed for adversarial attacks and defence against these attacks are many and diverse. CRT offers opportunities to leverage different computational methodologies to offer both micro and macro analyses of the reciprocal interaction between the defence and attack strategies. Macro analysis refers to the population-level analysis, while micro analysis refers to the study of individuals.

Following behavioural decision making theory, the effects of the personality of the red team members on $blue$ have been investigated using CRT, whereby the approach offered ways to use GA to evolve and identify the best personality characteristics of the $blue$ team required for a specific red behavioural space in a warfare game [34].

Moreover, areas such as computational game theory, epistemic logic and agent-based systems have been used to focus on modelling the action production of an AI. These methods can be used within a CRT system to model the agents.

However, this present study attempted to understand the observed and expressed actions of AIs and humans with no assumption of how the actions are produced. As such, the methodologies offered in the current paper can be extended to understanding observed behaviours from agents in both simulated and real environments alike.

In general, the responsiveness of the individual, in both humans and animals, determines their personalities [35]. The personality profile of an agent can be represented either indirectly through behavioural traits that will affect an agent's choice of an action, or directly through different categories of actions. The work presented in [34] explored different personality profiles for the $blue$ team in order to compete effectively against a red team with fixed personality profiles. Red teams with different personality profiles, including very aggressive, goal-oriented, defensive, cowardly and balanced were simulated. The actions of the red team were then influenced by the respective personality profile. Other personality traits included attraction or repulsion towards a healthy or injured friend. In the current study, we adopted a similar approach, whereby a 'goal-oriented' agent reflected the behaviour of a power agent whose focus was purely on the goal and less on exploring the environment. In contrast, the work in [34] was simulation-based, and therefore does not inform us of how a human would act when faced with an agent with these personality profiles.

The current study aimed at contrasting the behaviour of the human against the behaviour of the machine. The study achieved the aim by varying the quality of information an agent receives, the level of deception exhibited by an agent, and the effect of perceptual load.

The focus of the previous published work has been to establish the CRT framework and investigate noisy and deceptive information in the context of machines. In contrast, the focus of the current study is on contrasting the behaviours of the neuro-evolutionary-based red agents with the human-based red agents when they are positioned in the same CRT environment. In addition, we focused on the effect of perceptual loads on both types of red agent. To ensure a fair comparison, both types of red agent were positioned in the same CRT environment and the same analysis was conducted on both the human and machine data.

III. Materials and Methods

A. Apparatus

A controlled CRT environment allows the collection and analysis of actions taken by humans and AIs during competitive interactions. For this study, a simulation-based computational environment [12] was used to condition the CRT task and context. The environment provided the context in which two entities with conflicting objectives act: $blue$'s objective was to capture red, and red's objective was to avoid getting captured. The action sequences of the red and $blue$ agents could be viewed as the chase between a thief and a police. With the CRT framework, we could transform the action sequence space to a behavioural feature space. The framework extracted, transformed and represented the sequences of actions into behavioural data that enabled behavioural analysis. With the framework, we could further extract more meaningful information about the entities through characterising and mapping behavioural data. Given such characterised and mapped behaviour, our proposed framework catered for identifying points of interest in actions, and the

centre of mass of actions. These two metrics were essential for analysing the reasons for behaviour and behavioural changes in response to situations or stimuli within an environment.

Knowing that the $blue$ agent is watching, how would the red agent act to avoid the capture? For example, a thief who has information about the police's patrol route would probably not break into a house located on that route. What if the thief not only knows the police patrol route, but also knows the police's knowledge of the thief's location? Will the thief act differently or the same? We were particularly interested in studying whether red's knowledge of $blue$'s perception about red's position would influence and change red's behaviour in this reactive task. Therefore, two scenarios were simulated to study the effect of the red's knowledge of $blue$'s perception on its own behaviours. We labelled the scenario with the absence of red's knowledge of $blue$'s perception as *known-unknown*. On the other hand, the scenario in which red's knowledge of $blue$'s perception exists, was called *known-known*.

In the game, the strategies used by the $blue$ agent are affected by the quality of received information on the red agent's position (intelligence) and its own deceptive behaviours. The quality of intelligence is a function of two factors: the frequency of observing red and the sensorial noise encountered while observing red. The more frequent observations $blue$ can collect of red, the better. For example, more frequent observations can be used to filter out more sensorial noise. The less sensorial noise, the better the quality of the observations.

Deception is a complex concept in real-world situations. Chevalier-Skolnikoff [36] defines deception as 'the intentional sending of untrue signs to obtain some predetermined goal from a receiver's subsequent behaviour'. While Chevalier-Skolnikoff worked in a different context to ours, we adopted a similar perspective: that embedding false information within a communicated sequence is a form of deception.

We differentiated between random noise, known as white noise in classical communication research, and the use of a stochastic variable to model deception in this study. The former does not depend on a goal. The latter depends on the intention of an agent, and varies based on the agent's goal; thus strictly, it is not mathematically equivalent to white noise.

B. Independent Variables

Information conditions were varied by controlling three parameters: a) the level of noise, in terms of the frequency and accuracy of information sensed by the pre-programmed $blue$ agent; b) the level of deception, in terms of the frequency and magnitude of deviation from the desired actions taken by the pre-programmed $blue$ agent; and c) perception, in terms of the information availability to the red agent about $blue$'s perception of where red is. The first two (noise and deception) governed $blue$'s strategy. The third affected information availability to red. Parameters N_I and $\alpha^{(t)}$ controlled the frequency and noise, while N_D and $\zeta^{(t)}$ controlled the deception generated by $blue$.

Table I defines the levels of information quality used in the experiments. The value $N_I = 1$ indicates that the $blue$ agent received information about the red agent's position at each step of the simulation, while $N_I = 10$ indicates that the information was available at every 10th step.

TABLE I
THE COMBINATIONS OF N_I AND $\alpha^{(t)}$, GIVEN THAT THE DECEPTION EFFORT FROM THE $blue$ AGENT IS FIXED

Combination	N_I	$\alpha^{(t)}$	Description
1	1	0	Frequent and non-noisy information (Good quality)
2	1	$U(0,\ 20)$	Frequent and noisy information
3	10	0	Infrequent and non-noisy information
4	10	$U(0,\ 20)$	Infrequent and noisy information (Poor quality)

TABLE II
THE COMBINATIONS OF N_D AND $\zeta^{(t)}$, GIVEN THAT THE INFORMATION RECEIVED ABOUT THE red AGENT'S INTELLIGENCE IS FIXED

Combination	N_D	$\zeta^{(t)}$	Decription
1	1	0	Non-deceptive
2	10	$U(-15°,\ 15°)$	Less deceptive
3	10	$U(-30°,\ 30°)$	Moderately deceptive
4	5	$U(-15°,\ 15°)$	Moderately deceptive
5	5	$U(-30°,\ 30°)$	Highly deceptive

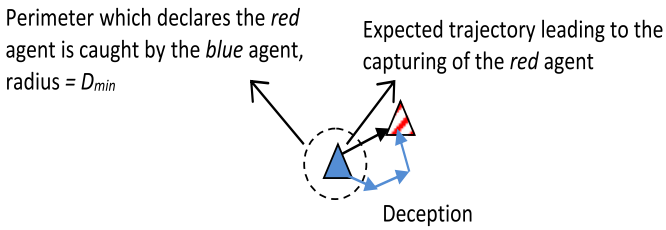

Fig. 1. Selection of maximum values of the uniform distribution for the generation of $\zeta^{(t)}$.

A uniform distribution, $U(0,\ 20)$, was used to generate $\alpha^{(t)}$. This represents the level of noise in $blue$'s sensing of the red agent's position. Noise was added by generating two random numbers in the given distribution range and adding them to the actual x and y coordinates. A range of 20 pixels was used to satisfy the $D_{\min}$ constraint required to trigger a capture event.

Table II defines the levels of deception. The first combination represents the scenario where the $blue$ agent always moves in the direction that it expects the other agent to be in, $\hat{P}_r(t)$. The other combinations represent scenarios in which the $blue$ agent deviates from its expected trajectory every N_D timed steps. The deviation from the trajectory depends on $\zeta^{(t)}$, which is sampled from a uniform distribution. Small ranges of deviation angles are selected to avoid side-tracking $blue$ from its intended goal (Fig. 1).

Four levels of noise and five levels of deception created 20 different situations for $blue$. Combined with the two different scenarios for red, there were 40 configurations in total.

In Table III, different noise and deception levels are labelled for ease of reference. The descriptions of the selected codomain are shown in Table III. For example, the combination of $N_I = 10$, $\alpha^{(t)} = U(0, 20)$ means the information was infrequent and noisy; thus, the quality of information was poor. On the other

TABLE III
DESCRIPTIVE LABELS FOR NOISE AND DECEPTION LEVELS

Definition	Parameter	Value	Description
Noise	Intel frequency	$N_I = 10$	Infrequent
		$N_I = 1$	Frequent
	Intel noise	$\alpha = 0$	Accurate
		$\alpha = U(0,\ 20)$	Noisy
Deception	Deception frequency	$N_D = 10$	Infrequent
		$N_D = 5$	Frequent
	Deception degree	$\zeta^{(t)} = U(-15^\circ,\ 15^\circ)$	Low
		$\zeta^{(t)} = U(-30^\circ,\ 30^\circ)$	High
	Combination	$N_D = 1, \zeta^{(t)} = 0$	No deception

hand, the combination of $N_D = 5$, $\zeta^{(t)} = U(-30^\circ,\ 30^\circ)$ means that the actions of $blue$ are highly deceptive.

The red in this framework was controlled either by a human agent or by an AI agent and provided the basis for comparing human and machine behaviour. Human subjects controlled the red agent through mouse clicks that pointed the agent to its next position in the $2D$ environment. The path taken by human players through these movements was recorded for analysis and for further comparison with the machine–based red.

C. Participants

A group of 34 postgraduate students (15 male, 19 female) from the local university campus volunteered to participate as the red agents in the experiments reported in this paper. The subjects were mature adults (30 ± 6.6 years old), recruited through an open advertisement on the university campus. The students were from diverse disciplines including the humanities, science, engineering, and information technology. Each subject participated in an orientation session before each experiment to familiarise themselves with the environment and the task.

Before the start of each experiment with a new subject, the researcher stepped him or her through the Participant Information Statement, answered the subject's questions, and collected the Consent Form to participate in the experiment. The subject was then shown the software game. The experiment was conducted only after the subject was satisfied that they had gained a sufficient understanding of the task and the skills needed to play the game.

At the conclusion of the experiments, we gave the subjects a questionnaire, which requested basic statistics on gender and age, and posed six questions. All subjects agreed that the briefing was sufficient. Most subjects ($31/34$) felt confident that they could play the game. Risk-appetite was almost normally distributed: some perceived that they were risk takers, and some felt the opposite, but most subjects sat at a moderate level of risk. All subjects felt that they were goal-driven in some sense. The majority of the subjects ($27/34$) used the score on the screen to inform their decisions. Only one subject tried to play in a random fashion, while 20 relied on re-planning their strategy after some moves.

Humans tend to act based on norms, which means they have knowledge or experience of how to react typically to certain known situations. In comparison, the machine needs to be trained beforehand to develop some sort of intelligence so that a fair comparison can be made between humans and machines. In the current study, the task was designed to require very minimum skills from participants. However, all participants had used computers extensively before this experiment and they were all comfortable with the use of the mouse and with the task itself.

There was no incentive involved in the experiments but chocolates were given at the end as a token of appreciation.

D. Procedure

One simulation run, or one game, consisted of a fixed number of timed steps, where both agents make equidistant moves according to their strategies at each timed step. Each game started with the initial separation distance between the red and $blue$ agents of 200 pixels. A simulation run could also terminate earlier, if the $blue$ agent captured the red agent.

Human reds played each configuration twice. The order of presenting the scenarios to the subjects had the potential to cause a learning effect and generate a bias in the data. Therefore, a counterbalanced experimental design was adopted to manage order bias. An exhaustive permutation of all factors would have generated an astronomical number (to the order of 20) of configurations. Instead, we used a Latin-squared design [37] to keep the experiment to a reasonable size, resulting in 40 configurations in total.

Each participant played a total of 80 games (40 different game configurations, each played twice with the *known-known* and *known-unknown* scenarios).

Participants played each single game continuously without a break. However, once a game had ended, the next game could only start by clicking a 'Continue' button displayed on the screen when the participant was ready. Therefore, participants were allowed to relax for a few seconds between each successive game. Between them, the 34 human participants played 2720 ($34 \times 40 \times 2$) games.

The experiments were then repeated using machine-based red agents. The computational red was controlled through a neuro–evolutionary model [38]. The chosen neuro–evolutionary controller was built upon autonomy and learning in artificial agents and was considered a suitable candidate to mimic these traits in humans. Neuro–evolution relies on evolving a population of neural networks through a GA [39].

A fixed and identical architecture was used to represent individual neural networks in the evolutionary framework. Each neural network consisted of a multi-layer perceptron of sigmoid units with a backpropagation algorithm. The learning rate for the network, η, was set at 0.2, and the momentum rate, γ, was set at 0. Each connection weight of the neural network lay in the interval $[-1, 1]$.

To further elaborate, the networks consisted of IN input, HN hidden and ON output neurons. The architecture is depicted in Fig. 2 where **IW** and **LW** refer to the connection weight matrices of $HN \times IN$ and $ON \times HN$ dimensions respectively between the input-hidden and hidden-output layers; while $\hat{\rho}^1$ with $IN \times 1$ dimensions and $\hat{\rho}^2$ with $ON \times 1$ dimensions refer to the vectors of bias units between the input-hidden and hidden-output layers.

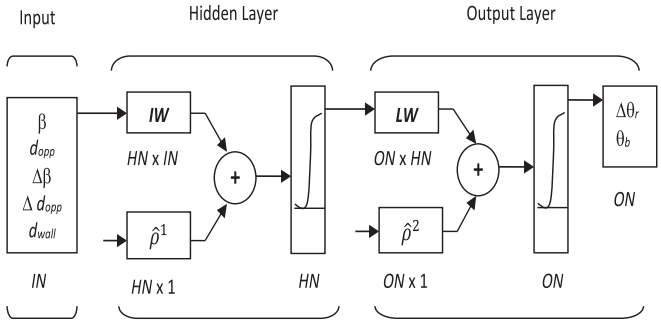

Fig. 2. Neural networks in the neuro-evolutionary-based model.

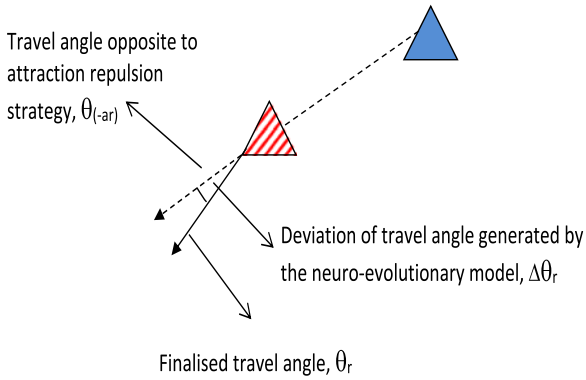

Fig. 3. Generation of travel angle for the red agent.

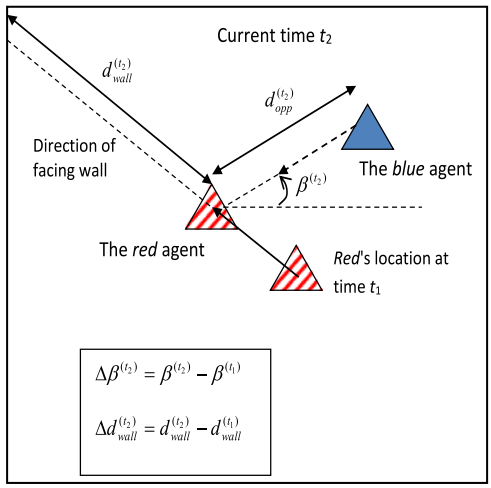

Fig. 4. Input variables for neural network.

The values of IN, HN and ON were set at 5, 7 and 2 neurons, respectively, in the experiments. The numbers of input and output neurons were determined in a straightforward way, as they were based on the numbers of input and output variables, respectively. Conversely, the number of hidden neurons was determined arbitrarily based on the need for a small network size, which required less processing time. Then, GA evolved the connection weights of the networks. It was found that the neural networks with evolved initial connection weights learned faster and better than those with initial random connection weights [40]. Further details about the neuro–evolution component of these experiments can be found in [12].

All six inputs to the neural networks were encoded as real numbers, and the GA used a real-number representation for the connection weights and biases. The inputs received by the neuro-evolutionary-based red in both scenarios are shown immediately below and a diagram for the inputs is given in Fig. 4:

1) $\beta^{(t)}$ – the relative angle between the $blue$ and red agents
2) $d_{\text{opp}}^{(t)}$ – the relative distance between the $blue$ and red agents at time t
3) $\Delta\beta^{(t)}$ – the relative change in β at time t
4) $\Delta d_{\text{opp}}^{(t)}$ – the relative change in d_{opp} at time t
5) $d_{wall}^{(t)}$ – the relative distance to the boundary of the simulation environment that the red agent faces at time t
6) $\alpha^{(t)}$ the level of noise in the information received by the $blue$ agent.

The outputs for the neuro–evolutionary-based model are shown as following:

1) $\Delta\theta_r^{(t)}$ – the deviation angle of the red agent
2) $\theta_b^{(t)}$ – the $blue$ agent's travel angle.

The red agent learned a strategy through a neuro-evolutionary process that always produced an angle, $\Delta\theta_r$, relative to the actual repulsion angle, $\theta_{(-ar)}$. The final travel angle for the red agent at each timed step t, $\theta_r^{(t)}$, was thus the sum of two angles (i.e. $\theta_r^{(t)} = \theta_{(-ar)}^{(t)} + \Delta\theta_r^{(t)}$), as illustrated in Fig. 3.

At this point, the planned travel angle of the red agent, $\theta_r^{(t)}$, was executed and the agent moved to a new location. At the same time, the $blue$ agent moved to a new location based on its own strategy. The actual travel angle of the $blue$ agent, $\acute{\theta}_b^{(t)}$, was compared with the angle predicted by the neuro–evolutionary-based model, $\theta_b^{(t)}$, and the difference was used to adjust the connection weights between network layers through backpropagation. Using the neural network as the control system, the red agent moved in the environment until the game was terminated. The fitness of each neural network was evaluated iteratively based on its performance (i.e., how successfully it escaped from being caught by the $blue$ agent) averaged over 10 repeated simulation runs. Once all neural networks in the population had been evaluated, GA was applied to create the next generation of neural networks.

The only difference between the input variables used by the neuro–evolutionary-based red agent in the $known - unknown$ and the $known - known$ scenario lies in the use of one extra input, $\alpha^{(t)}$.

Since the architecture of neural network is fixed, **IW**, **LW** , $\hat{\rho}^1$ and $\hat{\rho}^2$ were mapped into a vector of weights represented by a chromosome. $\hat{W}_{IW}$ refers to a vector of connection weights and bias units between the input-hidden layers, mapped as $\hat{W}_{IW} = \{\hat{w}_1^1, \hat{w}_2^1, \ldots, \hat{w}_{hn}^1, \ldots, \hat{w}_{HN}^1\}$. Similarly, $\hat{W}_{LW}$ refers to a vector of connection weights and bias units between the hidden-output layers, mapped as $\hat{W}_{LW} = \{\hat{w}_1^2, \hat{w}_2^2, \ldots, \hat{w}_{on}^2, \ldots, \hat{w}_{ON}^1\}$. The parameters in, hn and on with ranges $1 \leq in \leq IN$, $1 \leq hn \leq HN$, $1 \leq on \leq ON$ refer to the input, hidden and output layers' neurons.

GA randomly initialised a population of 100 individuals, each yielding a different set of connection weights for a neural network. As mentioned earlier, the network architecture and relevant learning parameters were fixed and identical for all individuals. All networks in the population were then evaluated for 10 games. The agent locations were initialised randomly in each simulation run, with the given initial separation distance of 200 pixels. Once all individuals had been evaluated using the fitness function in Equation 1, a new population was created using elitism and selective reproduction processes. The fitness function is proportional to the average number of time steps a red agent survived without being captured by the $blue$ agent within the lifetime of a simulation run as follows:

$$F = \frac{1}{N_G}\sum_{g=1}^{N_G} h_g \tag{1}$$

$$h_g = \begin{cases} 1 & \text{if } d_{\text{opp}} > D_{\min} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where h_g refers to the frequency of the red agent satisfying the constraint $(d_{\text{opp}} > D_{\min})$ in the g-th simulation run, with $1 \leq g \leq N_G$. This frequency, F, is accumulated for each step to crease an explicit fitness pressure to survive longer in the environment.

The fittest 10% of individuals (neural networks) from the population were copied to the next generation population. Next, new offspring were created iteratively and added to the new population. In each iteration, two parents were selected without replacement using the *binary tournament* selection method. An offspring was then generated using a *one-point* crossover and uniform mutation. During the mutation process, the mutation rate was used to perturb each weight in the chromosome using a randomly generated value according to a uniform distribution between -1 and 1. The process was repeated until the maximum population size is reached. The evaluation procedure explained above was then repeated for each individual neural network in the new population for 10 games, $N_G = 10$. The whole evolutionary cycle was repeated for 200 generations.

After 200 generations, the best individual in the final population was identified. The best evolved computational red agents each played another 10 simulated runs or games, repeated 30 times with different seeds, during which their manoeuvres were recorded for analysis. Thus, $40 \times 10 \times 30 = 12{,}000$ action sequences were recorded in total. Since the number of trials between human-based and neuro-evolutionary-based reds was different, statistical analysis based on hypothesis testing on the mean scores between the two types of red agents was carried out.

Both scenarios for human-based agents were simulated by different visualisation effects. The simulations involving humans for both scenarios are shown in Figs. 5 and 6. In the *known-unknown* scenario (Fig. 5) the human player did not know the $blue$ agent's perception of the red agent's position. In the *known-known* scenario (Fig. 6), red could see $blue$'s perception of red's position. This implies that the human player saw the level of noise, $\alpha^{(t)}$, in the information received by the

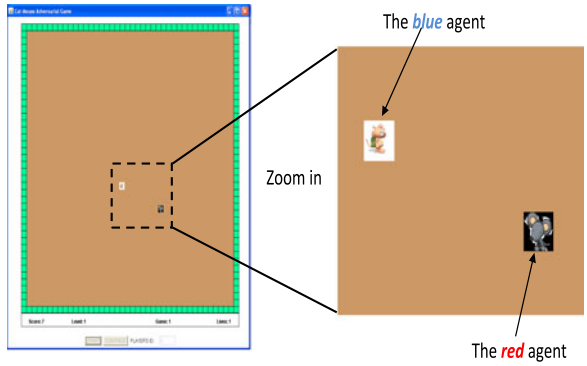

Fig. 5. Illustration of the *known-unknown* scenario.

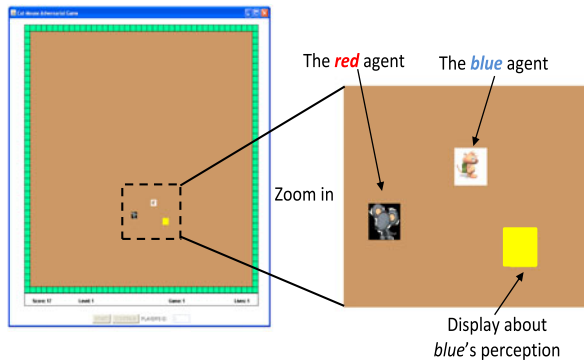

Fig. 6. Illustration of the *known-known* scenario.

$blue$ agent on the screen. In contrast, the level of noise, $\alpha^{(t)}$ in the information was an unknown to the red agent in the *known-unknown* scenario. The information available to the human was similar to the inputs of the neuro-evolutionary-based agents. However, the form to communicate the information to the human was clearly different. Visualisation was used to communicate the information to the human, while the information was directly fed as inputs to the machine. For the simulations involving humans, the scores of each game were displayed at the bottom of the screen as a feedback for the human to assess their own performance from one game to another.

E. Data Analysis

In the game, $blue$ was considered to have captured red if the distance between them fell below a threshold of $D_{\min}$. This threshold was set at 20 pixels, based on the resolution of the screen for a human player. Red agents were given a score of 1 at each timed step if they were not caught by the $blue$ agent. Reaction time ('RT') captured the average inter-step time of human reds. This was measured by the inter-click event time of two clicks.

The differences in terms of scores and reaction times were analysed using a t-test. A t-test was carried out to evaluate whether or not mean scores obtained by neuro–evolutionary-based red and human-based red for the same configuration and scenario were equal. Forty pairwise (20 configurations $\times$

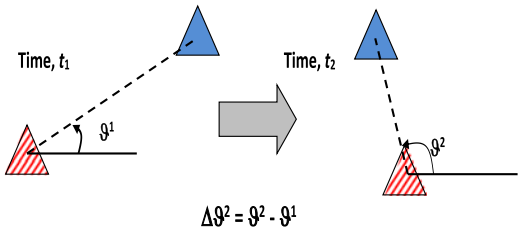

Fig. 7. Calculation for the travel angle change, $\Delta\vartheta^t$.

2 scenarios) t-tests were conducted between the two different setups of red agents. If the null hypothesis for having equal means was rejected, a further hypothesis test based on left-tail or right-tail was carried out.

In trajectory visualisation, the agents' action sequences or movements were plotted in the 2D simulation environment. The trajectories for the red and $blue$ agents – starting at solid dots and ending at the crosses – were represented by red solid and $blue$ dash-dotted lines respectively.

While trajectory visualisation is an intuitive analysis approach, to analyse a large number of simulations, the process becomes time consuming and cognitively demanding on a human operator. The methodology for action sequence analysis conducted in the current study aimed to capture the essence of the strategies of an agent across a number of simulation runs. This was done by conducting a frequency distribution analysis of the relative 'heading change' feature, recorded as the difference between the two agents' travel angles at each timed step.

Since each simulation run terminated at different times, T, we had a vector of the relative moving angles of red, ϑ^t. The vector representing a sequence of ϑ^t, was denoted as $\hat{\vartheta} = \{\vartheta^1, \vartheta^2, \ldots, \vartheta^t, \ldots, \vartheta^T\}$. We were able to obtain a sequence of changes in $\hat{\vartheta}$ and this vector was denoted as $\Delta\hat{\vartheta} = \{\Delta\vartheta^2, \Delta\vartheta^3, \ldots, \Delta\vartheta^t, \ldots, \Delta\vartheta^T\}$. Here $\Delta\vartheta^t$ refers to $\vartheta^t - \vartheta^{t-1}$, as illustrated in Fig. 7. To perform statistical analysis, each single sequence of ϑ and $\Delta\vartheta$ was viewed as a sample. Finally, the frequency distributions were plotted across all samples and a central-tendency-based curve fitting approach was used to summarise the overall pattern.

While the 'heading change' feature used in the frequency analysis provided a means to analyse agents' strategies at every timed step, the action sub-sequence analysis captured movement patterns on a larger timescale using a windowing concept.

In a nutshell, the sub-sequence similarity analysis consisted of dividing action sequences across different simulation runs into an equal number of sub-sequences, computing velocity and acceleration vectors for each of these sub-sequences, and measuring the similarity between behaviours (action sequences) using fuzzy c-means (FCM) unsupervised clustering. FCM allows the instances to belong to several clusters simultaneously with different degrees of membership in each cluster [41]. We considered this to be a better technique for this analysis, owing to the complexity of the actions produced by the agents.

The sub-sequences were extracted by dividing the action sequences obtained from each simulation run into an equal number of windows (Fig. 8 demonstrates this concept pictorially).

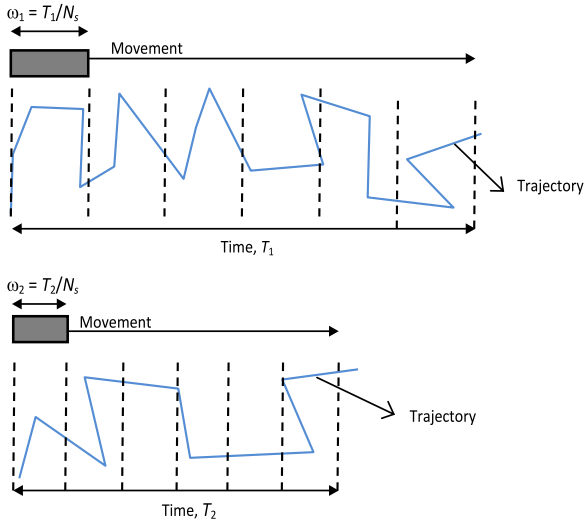

Fig. 8. Two examples of action sequences divided into equal numbers of sub-sequences. Each action sequence has a different length (referred as T_1 and T_2) which leads to different sub-sequence lengths (referred to as ω_1 and ω_2).

Since each simulation run could last for a different number of timed steps, the number of sub-sequences, N_S, was chosen based on the smallest sequence observed across all simulation runs. Sub-sequences were then extracted from each sequence as $\omega_{i_j} = len(A_i)/N_s$, where $\omega_{i_j}, j \in (1, N_s)$ referred to the jth sub-sequence of the ith action sequence A_i and $len(A_i)$ corresponded to the number of timed steps in A_i.

The number of sub-sequences in each action sequence was chosen based on the smallest sequence size. Based on the samples of action sequences in the experiments involving humans and neuro-evolution, the smallest sequence size was 10. At each window, for example, window 1 (w1), we extracted velocity information. Therefore, there were 10 velocities for each action sequence. The total feature vector size was 19, with the first 10 features referring to velocity, $V_{r\ rel\ b}$, while the remaining nine features referred to acceleration, $\Delta V_{r\ rel\ b}$. Details of the calculation of $V_{r\ rel\ b}$ and $\Delta V_{r\ rel\ b}$ can be found in [12].

In our research, each data instance can be viewed as a point in a time series with an equal window length of 19, and as a behavioural sequence of the red agents in different conditions. The 19 features in a time series were used as the input vector for clustering the data based on similarities. In this study, clustering was used to categorise the data (behavioural sequences) into pre-defined clusters (cluster size) based on the input variables to identify possible behavioural patterns of the red agents in different conditions. To perform clustering, the different time series need to be of equal length (i.e., the same number of input variables). An appropriate number of clusters should be determined based on the adopted cluster validity method. The internal criteria-based method was chosen to evaluate the appropriate number of clusters (between 2 and 10) in our analysis. At the conclusion of the clustering, each cluster centroid was viewed as a representative pattern of each cluster. Therefore, clustering helped to summarise the behaviour patterns shown by both groups red agents in different conditions,

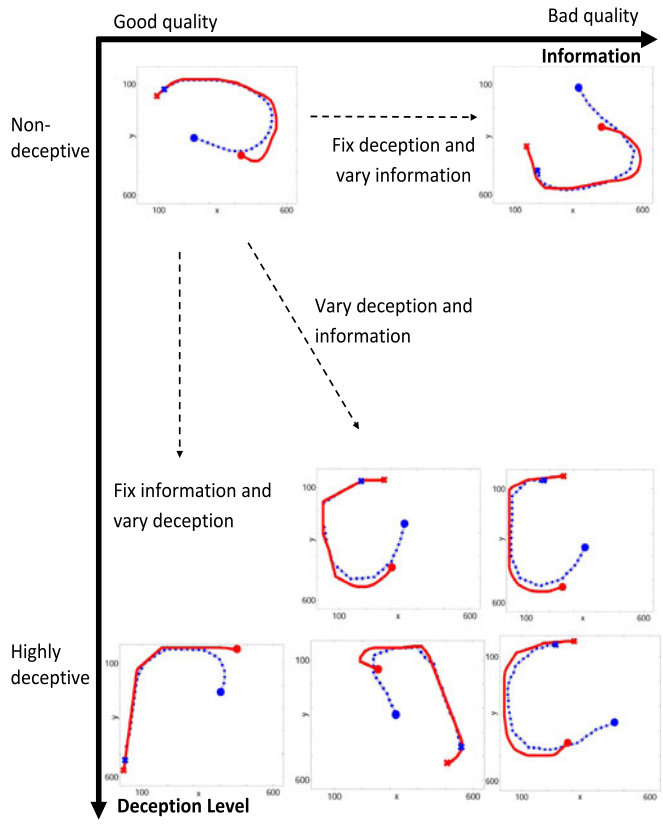

Fig. 9. Trajectories shown by the $blue$ agent and human for different combinations of information and deception in the *known-unknown* scenario. Both x-y axes in all inset graphs are in the range [0, 600].

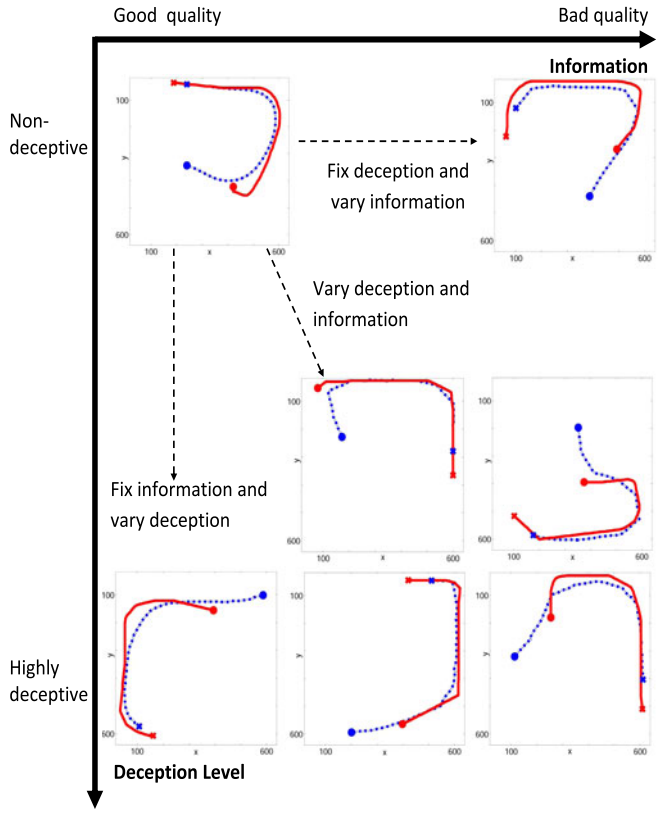

Fig. 10. Trajectories shown by the $blue$ agent and human for different combinations of information and deception in the *known-known* scenario. Both x-y axes in all inset graphs are in the range [0, 600].

whereby the characteristics of the time series were taken into consideration.

IV. Results

For the t-test, the significance level was set to be 0.05 ($\alpha = 0.05$) to investigate the difference in terms of scores and reaction time in both scenarios. The rest of this section details the different findings.

A. Trajectory Visualisation

Figs. 9 and 10 show the trajectories across different experimental configurations for the same human red player in the *known-unknown* and *known-known* scenarios, respectively. Meanwhile, Figs. 11 and 12 show the trajectories of the neuro-evolutionary red agent for the same starting conditions (same seeded runs) in both scenarios. In these figures, the quality of information is shown to deteriorate as we move from left to right, while the deception level is shown to increase as we move from top to bottom.

The trajectories shown in Figs. 9 and 10 can be viewed as a good representation of the common trajectories of the 34 human players. This is because most human manoeuvring patterns in both scenarios look similar, consisting of a combination of simple curve and straight-line trajectories across different configurations. Human players seemed to adopt only a single strategy, regardless of information accuracy and deception. This finding could be closely related to personal perception and self-perception concepts used in social cognition, whereby aspects of perception may influence the mental states of humans during their interaction with different environments [42]. Humans tend to adapt to different environments by relying on the knowledge represented in their memories. How human participants react in the games is related closely to how humans perceive the interaction between a cat and a rat in the real world, whereby we always observe that a rat runs in the opposite direction whenever it encounters a cat. It seems that humans relied on these stored experiences and retrieved them to display similar behaviours to those they had observed in the real world.

Since the neuro–evolutionary-based agents played 10 simulated runs or games, repeated 30 times with different seeds for each configuration, there were $30 \times 10 = 300$ action sequences. The performances of the neuro–evolutionary-based agents were different across trials, however, it was impossible to display all 300 action sequences in a graph. Therefore, only a few interesting patterns (those with action sequences that differed from those of human-based agents.) were selected across trials for display but some of the patterns were found repeatedly in different trials.

Unlike human agents, the neuro-evolutionary red agents' strategies showed more variation under different combinations of intelligence and deception. The sample trajectories presented

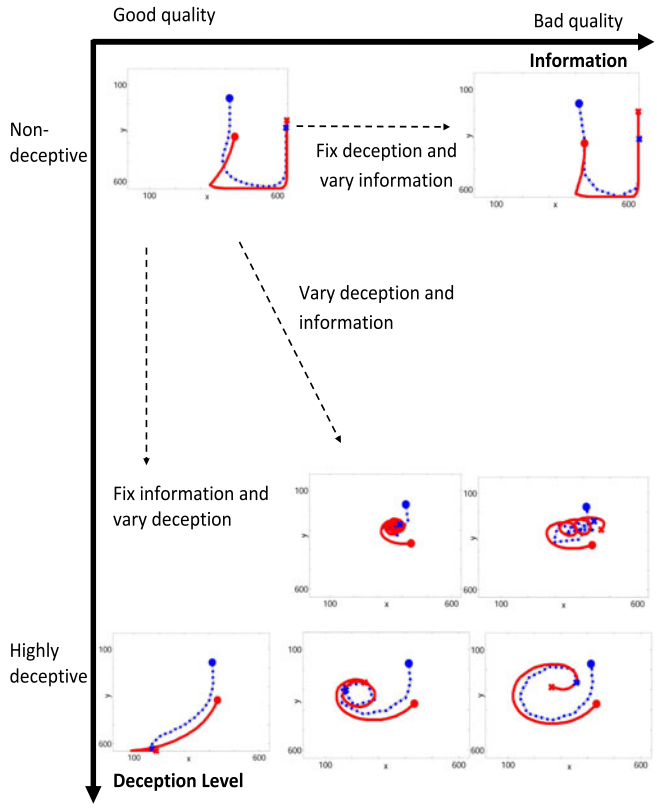

Fig. 11. Trajectories shown by the $blue$ and neuro–evolutionary red agents for different combinations of information and deception in the *known-unknown* scenario. Both x-y axes in all inset graphs are in the range [0, 600].

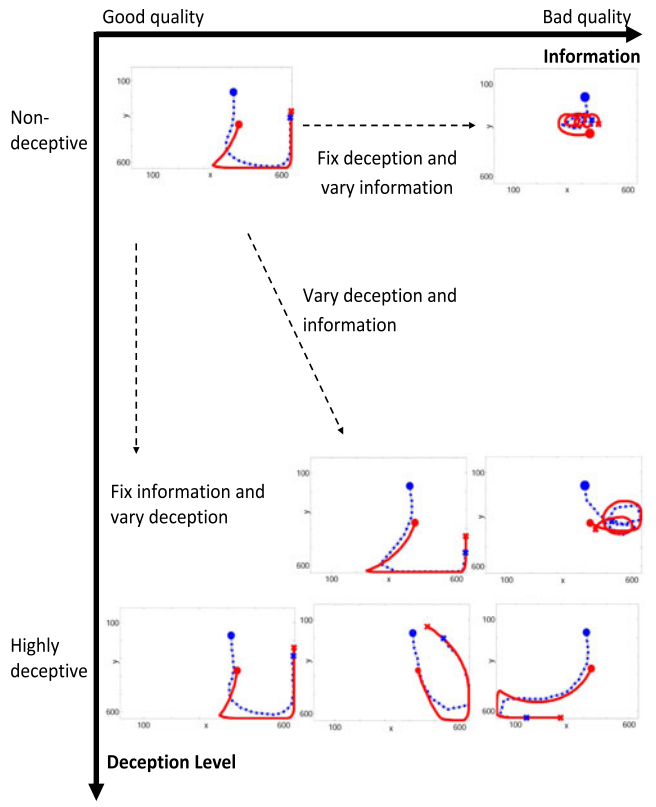

Fig. 12. Trajectories shown by the $blue$ and neuro–evolutionary red agents for different combinations of information and deception in the *known-known* scenario. Both x-y axes in all inset graphs are in the range [0, 600].

in Figs. 11 and 12 show more complicated and interesting patterns than do the human trajectories. Simple visual inspections seem to indicate that the neuro-evolutionary red changed its strategies in response to the combinations of intelligence and deception. The neuro-evolutionary red showed similar manoeuvring patterns across different levels of deception in both scenarios when the $blue$ agent received frequent and accurate information. The manoeuvring patterns followed simple curve and straight line trajectories. As the frequency and quality of received information changed from being accurate and frequent to noisy and infrequent, we can see that most of the trajectories became more curvy, implying a tendency to correct actions in the presence of noise. The most interesting behaviours seemed to emerge in the extreme situation of infrequent and noisy information, where most of the trajectories took circular spiral shapes with what appeared to be a chaotic motion.

In summary, the red agent using neuro–evolution appeared to be more diverse in adopting strategies to avoid the $blue$ agent compared with the human players. Further, the diversity of the neuro-evolutionary red agent increased with the increase of information deterioration and deception. These interesting trajectories could indicate that the neuro-evolutionary red agent was actually projecting where it thought $blue$ thought red was, based on the observations of the $blue$ agent's movements.

B. Action Frequency Distribution

The supplementary materials presented in Appendix A provide a detailed analysis of the frequency distribution of both human and machine actions in both experimental scenarios.

This study has demonstrated that humans were not influenced by the extra information provided about $blue$'s perception in the *known-known* scenario. Moreover, evidence exists to indicate that human players repeated the experience every time without transfer of knowledge from one game to another. This is further supported by an analysis of the time-to-decision (which is equivalent in our experimental setup to reaction time) showing no difference between the two scenarios; which further indicates that humans did not process the extra information provided about $blue$'s perception.

The neuro-evolutionary-based agents consistently performed better than humans. Unlike the humans, the neuro-evolutionary-based red agent specialised through knowledge transfer from one generation to another during evolution.

C. Sub-Sequence Similarity Analysis

The cluster validity indices were used to determine an appropriate number of clusters when clustering the sub-sequences of both agents. The number of clusters determined the possible homogeneous groups that the underlying data points belonged to. With respect to our research, the number of clusters

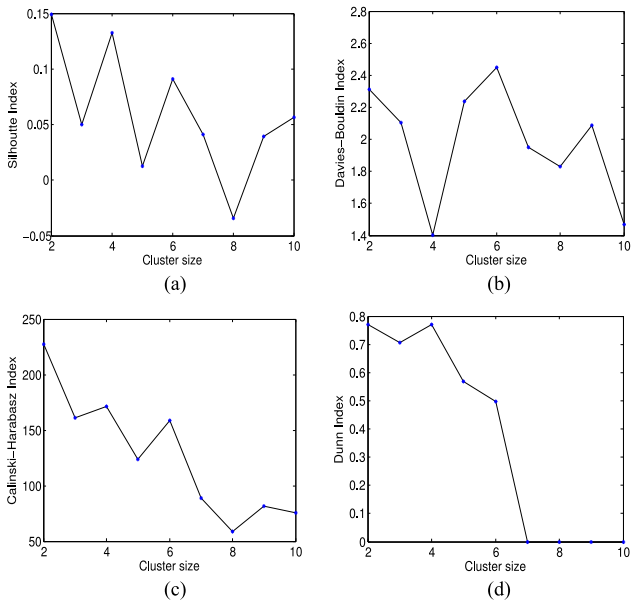

Fig. 13. Cluster validity indices for the action sub-sequence data set for humans in the *known-unknown* scenario. (a) Silhoutte Index. (b) Davies-Bouldin Index. (c) Calinski-Harabasz Index. (d) Dunn Index.

TABLE IV
RANKING OF NUMBER OF CLUSTERS BASED ON CLUSTER VALIDITY INDICES FOR HUMANS IN THE *Known-Unknown* SCENARIO

Index	Number of clusters								
	2	3	4	5	6	7	8	9	10
Silhoutte	1	7	2	8	3	5	9	6	4
Davis-Bouldin	8	6	1	7	9	4	3	5	2
Calinski-Harabasz	1	4	2	5	3	6	9	7	8
Dunn	2	3	1	4	5	6	7	8	9
Average	3.00	5.00	**1.50**	6.00	5.00	5.25	7.00	6.50	5.75

determined the possible types of behaviours that the agents show. For example, two clusters meant that the action sequences would be categorised into two main groups, in which each centroid of the group represented the behaviour type. Action sequences that belonged to the same group could be considered as similar behaviours to those from another group. The results show the same number of clusters in all scenarios except the *known-unknown* scenario involving the neuro-evolutionary red agent, whereby there was a disagreement in determining the number of clusters. For example, Fig. 13 illustrates that the Silhoutte and Calinski-Harabasz indices recommended two clusters, the Dunn index suggested four clusters and the Davis-Bouldin index suggested four clusters for the case of involving humans in the *known-unknown* scenario. To resolve this disagreement, where the cluster validity indices showed different preferences for the number of clusters, we ranked the clusters.

The ranking procedure worked such that a lower rank was given to the cluster size associated with better indices and vice versa. After that, the appropriate number of clusters was determined based on the average rank, as shown in Table IV. The

TABLE V
NUMBER OF CLUSTERS BASED ON THE CLUSTER VALIDITY INDICES

Red	Scenario	Number of Clusters
Humans	Known-Unknown	2
	known-known	4
Neuro–evolution	Known-Unknown	2
	known-known	2

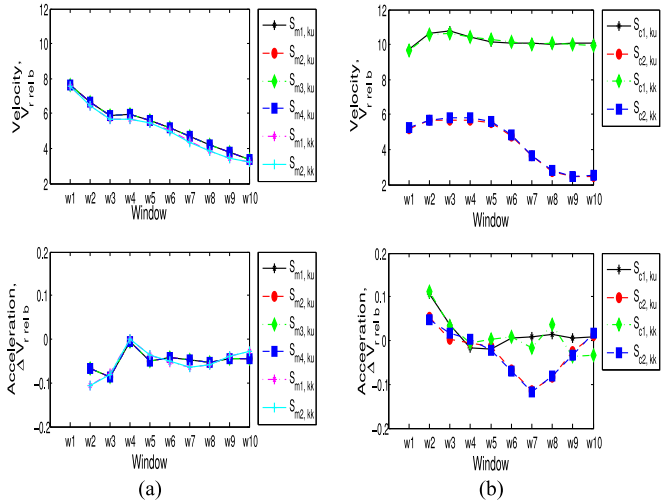

Fig. 14. $V_{r\,rel\,b}$ and $\Delta V_{r\,rel\,b}$ for the action sequences for the human and neuro–evolutionary red agents. (a) The humans. (b) The neuro.evolutionary red agent.

results in Table IV suggest that the appropriate number of clusters is four with an average ranking value of 1.50 for humans in the *known-unknown* scenario. The cluster validity indices for the remaining experiments are summarised in Table V.

Based on the number of clusters shown in Table V, the FCM clustering method was applied to the data set built from the velocity or acceleration features of sub-sequences. For ease of visualisation, the cluster centroid values for both velocity and acceleration features are shown in Fig. 14(a) for the human red agents and Fig. 14(b) for the neuro-evolutionary red agents.

As shown in Fig. 14(a), it is surprising to observe that only a single strategy was adopted by the humans in both scenarios. Even though the cluster validity indices indicated four and two strategies in the *known-unknown* and *known-known* scenarios respectively, the strategies overlapped with each other, that is, $S_{m1,ku} - S_{m4,ku}$ in the *known-unknown* scenario and $S_{m1,kk} - S_{m2,kk}$ in the *known-known* scenario. The difference between the strategies of both scenarios rests in $\Delta V_{r\,rel\,b}$, but the difference is marginal. For the neuro-evolutionary–based red agent, its action sequences in both scenarios can be categorised into two obvious strategies as shown in Fig. 14(b), that is, $S_{c1,ku} - S_{c2,ku}$ in the *known-unknown* scenario and $S_{c1,kk} - S_{c2,kk}$ in the *known-known* scenario.

For each game configuration, there were 300 sub-sequences built from the velocity or acceleration features of the neuro-evolutionary-based reds, and 68 sub-sequences from the velocity or acceleration of the human reds. This makes direct

TABLE VI
THE PERCENTAGE OF TRAJECTORIES IN THE *Known-Known* AND *Known-Unknown* SCENARIO THAT FALL WITHIN EACH CLUSTER

Configuration				*known-known*		*known-unknown*	
N_I	$\alpha^{(t)}$	N_D	$\zeta^{(t)}$	Cluster 1	Cluster 2	Cluster 1	Cluster 2
		1	0	2.7	97.3	2.0	98.0
		5	$U(-15°, 15°)$	4.0	96.0	4.3	95.7
1	0	5	$U(-30°, 30°)$	5.7	94.3	4.3	95.7
		10	$U(-15°, 15°)$	4.0	96.0	6.7	93.3
		10	$U(-30°, 30°)$	6.0	94.0	6.0	94.0
		1	0	3.7	96.3	8.0	92.0
		5	$U(-15°, 15°)$	5.3	94.7	3.7	96.3
1	$U(0, 20)$	5	$U(-30°, 30°)$	3.0	97.0	7.7	92.3
		10	$U(-15°, 15°)$	5.7	94.3	3.7	96.3
		10	$U(-30°, 30°)$	6.0	94.0	3.3	96.7
		1	0	71.3	28.7	66.0	34.0
		5	$U(-15°, 15°)$	78.0	22.0	53.0	47.0
10	0	5	$U(-30°, 30°)$	72.3	27.7	59.7	40.3
		10	$U(-15°, 15°)$	72.0	28.0	78.0	22.0
		10	$U(-30°, 30°)$	61.0	39.0	81.7	18.3
		1	0	55.0	45.0	50.0	50.0
		5	$U(-15°, 15°)$	58.0	42.0	60.7	39.3
10	$U(0, 20)$	5	$U(-30°, 30°)$	47.7	52.3	63.7	36.3
		10	$U(-15°, 15°)$	61.0	39.0	56.3	43.7
		10	$U(-30°, 30°)$	249.0	51.0	48.3	51.7

comparison difficult. Therefore, the sub-sequences of both neuro-evolutionary-based and human reds were categorised into a number of representative groups (or clusters). The centers of the formed clusters became the representative clusters. A comparison was then made between the cluster centroid of the neuro-evolutionary-based and human reds, as shown in Table V.

By scrutinising Fig. 14(a) and (b), one can observe that the evolved agents have two cluster centroids, while the cluster centroid for humans fell mostly in between these two cluster centroids of the evolved agents. We recognised that the neuro-evolutionary–based red agents were trained with different configurations. We therefore investigated whether the diversity of strategies was caused by the training regime, or whether it was an inherent characteristic of neural-based training.

To answer the above question, we calculated the percentage of trajectories from each configuration that existed in each cluster, as shown in Table VI.

One can distil two important observations from Table VI. First, the majority of the generated trajectories when information on $blue$ was frequent belonged to Cluster 2. In fact, the composition of Clusters 1 and 2 in this case did not seem to differ between the *known-known* and *known-unknown* scenarios.

This situation changed when information on $blue$ was infrequent. In this particular case, when noise in the communication channel was absent and the frequency of red's deceptive actions was low, the two clusters shared the generated trajectories in the *known-unknown* scenario almost equally, while Cluster 1 hosted most of the trajectories in the *known-known* scenario. In the remaining cases, the trajectories in both scenarios tended towards falling into Cluster 1.

The above discussion reveals that diversity in trajectories exists regardless of the training configuration adopted. Nevertheless, the distribution of these trajectories can be different

TABLE VII
THE RESULTS OF t-TEST FOR THE MEAN SCORES BETWEEN STRATEGIES 1 AND 2 IN THE *Known-Unknown* AND *Known-Known* SCENARIOS FOR CRT

Scenario	Hypotheses	df	t-value	p-value	Reject H_0
known-unknown	$H_0: \mu_{s1,ku} \geq \mu_{s2,ku}$ $H_1: \mu_{s1,ku} < \mu_{s2,ku}$	5998	-10.436	0.00	Yes
known-known	$H_0: \mu_{s1,kk} \geq \mu_{s2,kk}$ $H_1: \mu_{s1,kk} < \mu_{s2,kk}$	5998	-15.604	0.00	Yes

TABLE VIII
SCORES AND FREQUENCIES FOR STRATEGIES 1 AND 2 IN THE *Known-Unknown* AND *Known-Known* SCENARIOS FOR CRT

Strategy	*known-unknown*		*known-known*	
	Score, $\mu \pm \sigma$	Frequency	Score, $\mu \pm \sigma$	Frequency
Strategy 1	93.2 ± 20.5	2001 (33.4%)	90.3 ± 24.1	2014 (33.6%)
Strategy 2	97.3 ± 9.8	3999 (66.6%)	97.2 ± 9.9	3986 (66.4%)

and is primarily affected by information on $blue$. This raises a second question: could neuro-evolutionary-based reds trained using frequent information on $blue$ with no deception survive the harsh environment of less frequent information about $blue$ and high deception? If, indeed, diversity exists in each training configuration, then the answer to this question should be 'yes'. We will answer this question in Section IV-D.

To investigate the difference in scores among these strategies, a t-test was carried out with $\alpha = 0.05$ to evaluate the null hypothesis that the samples in one of the strategies have greater or equal means compared with the other strategy. The mean scores for strategies 1 and 2 in the *known-unknown* scenarios were denoted as $\mu_{s1,ku}$ and $\mu_{s2,ku}$, while they were denoted as $\mu_{s1,kk}$ and $\mu_{s2,kk}$ in the *known-known* scenario. The results of the t-test are shown in Table VII. The scores for Strategy 1 were significantly lower than those of Strategy 2. However, the scores for both strategies are still high, with both means exceeding 90%. In addition, the results in Table VIII demonstrate that the frequency of the better strategy (Strategy 2) was higher than Strategy 1. In general, most of the strategies generated by evolution are considered good.

The sub-sequence similarity analysis suggests that strategies for humans and machines were quite different from each other. Even though both forms of red agents were exposed to the same task, their behaviours were different and their action preferences were influenced differently by information and deception.

D. Different Train and Test Configurations

Even though our experiment focused on a comparison between human and machine behaviours in the same adversarial environment, we thought it would be interesting to compare the red agent trained under good quality, non-deceptive information and then tested under highly noisy, highly deceptive information. Therefore, the configuration was repeated by using 30 different seeds. The red was able to evade being captured most

TABLE IX
SCORES AND FREQUENCY OF SURVIVAL OBTAINED BY THE red AGENT WHEN TRAINED UNDER GOOD QUALITY, NON-DECEPTIVE INFORMATION, AND TESTED UNDER HIGHLY NOISY, HIGHLY DECEPTIVE INFORMATION

Scenario	Score ($\mu \pm \sigma$)	Survival Frequency	Non-Survival Frequency
known-unknown	98.72 ± 8.69	293 (97.7%)	7 (2.3%)
known-known	97.58 ± 12.08	288 (96.0%)	12 (4.0%)

of the time, as shown by the high scores and survival frequencies of the red agent in Table IX.

V. CONCLUSION

This paper has presented a study focused on understanding the behaviours of humans and artificial agents through the lens of CRT. A CRT-based framework was used to measure, analyse and compare the behaviours of humans and machines. The study reveals that humans tend to behave in a similar manner and, when focusing on a task, they miss information that is readily available to them, perhaps because of perceptual load.

Extrapolating from our findings, strategies evolved by computers could complement humans' evolved strategies. The study has demonstrated that the strategies evolved by AI are diverse and do not resemble humans' strategies. This offers an opportunity to complement human strategies with those diverse AI-based strategies.

The implication of our research is that it is possible that human-only RT exercises do not consider these diverse AI-based strategies, causing an internal threat to the effectiveness of the exercise. The findings suggest that humans may ignore useful information under tight time constraints and high perceptual loads. The results suggest that a combination of humans and machines could possibly develop better strategies. In other words, the behaviour of the machine could augment human actions with different types of strategies from those exhibited by the human. Therefore, it is important to continue to characterise human and machine behaviours, to evaluate the pros and cons, and to design strategies for humans and machines to work together symbiotically.

ACKNOWLEDGMENT

This work is funded by the Australian Research Council Discovery Grant number DP140102590 and the Universiti Sains Malaysia (USM) Short Term Grant number 304/PCGSS/6313128.

REFERENCES

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Egnlewood Cliffs, NJ, USA: Prentice-Hall, 1995.
[2] J. Martínez-Miranda, and A. Aldea, "Emotions in human and artificial intelligence," *Comput. Human Behav.*, vol. 21, no. 2, pp. 323–341, 2005.
[3] J.-M. Hoc, "From human–machine interaction to human–machine cooperation," *Ergonomics*, vol. 43, no. 7, pp. 833–843, 2000.
[4] S. B. Griffith, *Sun Tzu—The Art of War*. London, U.K.: Oxford Univ. Press, 1963.
[5] DOD, "Defense science board task force on the role and status of DoD red teaming activities," Office Under Secretary Defense Acquisition, Technol, Logistics, Washington, DC, USA, Tech. Rep. Unclassified Rep. 20301-3140, 2003.
[6] J. F. Sandoz, "Red teaming: A means to military transformation," Inst. Defense Anal., Alexandria, VA, USA, IDA Paper P-3580, 2001.
[7] H. A. Abbass, *Computational Red Teaming: Risk Analytics of Big-Data-to-Decisions Intelligent Systems*. Berlin, Germany: Springer, 2015.
[8] H. A. Abbass, A. Bender, S. Gaidow, and P. Whitbread, "Computational red teaming: Past, present and future," *IEEE Comput. Intell. Mag.*, vol. 6, no. 1, pp. 30–42, Feb. 2011.
[9] S. Lima, "Putting predators back into behavioral predator–prey interactions," *Trends Ecol. Evol.*, vol. 17, no. 2, pp. 70–75, 2002.
[10] P. Riley and M. Veloso, "On behavior classification in adversarial environments," in *Distributed Autonomous Robotic Systems 4*. Berlin, Germany: Springer, 2000, pp. 371–380.
[11] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, 2010.
[12] S. L. Wang, K. Shafi, C. Lokan, and H. A. Abbass, "An agent based model to simulate and analyse behaviour under noisy and deceptive information," *Adaptive Behav.*, vol. 21, no. 2, pp. 96–117, 2013.
[13] H. J. Einhorn and R. M. Hogarth, "Behavioral decision theory: Processes of judgement and choice," *Annu. Rev. Psychol.*, vol. 32, no. 1, pp. 53–88, 1981.
[14] H. A. Kautz, "A formal theory of plan recognition," Ph.D. dissertation, Bell Lab., Murray Hill, NJ, USA, 1987.
[15] S. Carberry, "Techniques for plan recognition," in *User Modeling and User-Adapted Interaction*, vol. 11. Berlin, Germany: Springer, 2001, nos. 1/2, pp. 31–48.
[16] E. Santos Jr, D. Li, and X. Yuan, "On deception detection in multi-agent systems and deception intent," *Proc. SPIE*, vol. 6965, 2008, Art. no. 696502.
[17] J. Krozel and D. Andrisani, "Intent inference with path prediction," *J. Guid., Control, Dyn.*, vol. 29, no. 2, pp. 225–236, 2006.
[18] L. W. Wagenhals, T. Reid, R. J. Smillie, and A. H. Levis, "Course of action analysis for coalition operations," DTIC, Fort Belvoir, VA, USA, Tech. Rep., 2001.
[19] D. A. Gilmour, J. P. Hanna, W. E. McKeever Jr., and M. J. Walter, "Real-time course of action analysis," DTIC, Fort Belvoir, VA, USA, Tech. Rep. AFRL-IF-RS-TP-2006-6, 2006.
[20] J. W. Payne, J. R. Bettman, and E. J. Johnson, "Behavioral decision research: A constructive processing perspective," *Annu. Rev. Psychol.*, vol. 43, no. 1, pp. 87–131, 1992.
[21] H. A. Abbass, S. Alam, and A. Bender, "MEBRA: Multiobjective evolutionary-based risk assessment," *IEEE Comput. Intell. Mag.*, vol. 4, no. 3, pp. 29–36, Aug. 2009.
[22] H. A. Abbass and A. Bender, "The Pareto operating curve for risk minimization," *Artif. Life Robot.*, vol. 14, no. 4, pp. 449–452, 2009.
[23] S. Alam *et al.*"Discovering delay patterns in arrival traffic with dynamic continuous descent approaches using co-evolutionary red teaming, "*Air Traffic Control Quart.*, vol. 20, no. 1, pp. 1–10, 2012.
[24] G. H. Burgin, "An adaptive maneuvering logic computer program for the simulation of one-on-one air-to-air combat: Program description," NASA, Washington, DC, USA, Tech. Rep. NASA-CR-2582, 1975.
[25] C. Hayes and J. Schlabach, "FOX-GA: A planning support tool for assisting military planners in a dynamic and uncertain environment, " in, *Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments*. Palo Alto, CA, USA: AAAI, pp. 21–26, 1998.
[26] J. L. Schlabach, C. C. Hayes, and D. E. Goldberg, "Fox-ga: A genetic algorithm for generating and analyzing battlefield courses of action," *Evol. Comput.*, vol. 7, no. 1, pp. 45–68, 1999.
[27] P. Hingston and M. Preuss, "Red teaming with coevolution," in *Proc. IEEE Congr. Evol. Comput.*, 2011, pp. 1155–1163.
[28] D. Lowd and C. Meek, "Adversarial learning," in *Proc. Eleventh ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2005, pp. 641–647.
[29] J. Newsome, B. Karp, and D. Song, "Paragraph: Thwarting signature learning by training maliciously," in *Recent Advances in Intrusion Detection*. Berlin, Germany: Springer, 2006, pp. 81–105.
[30] A. Veloso and W. Meira Jr, "Lazy associative classification for content-based spam detection," in *Proc. Fourth Latin Amer. Web Congr.*, 2006, pp. 154–161.
[31] Z. Jorgensen, Y. Zhou, and M. Inge, "A multiple instance learning strategy for combating good word attacks on spam filters," *J. Mach. Learn. Res.*, vol. 9, pp. 1115–1146, 2008.

[32] B. Nelson *et al.*, "Misleading a learner: Co-opting your spam filter," in *Machine Learning in Cyber Trust*. New York, NY, USA: Springer, 2009, pp. 17–51.
[33] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in *Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY USA: ACM, 2004, pp. 99–108.
[34] A. Yang, H. A. Abbass, and R. Sarker, "Characterizing warfare in red teaming," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 2, pp. 268–285, Apr. 2006.
[35] M. Wolf, G. S. van Doorn, and F. J. Weissing, "Evolutionary emergence of responsive and unresponsive personalities," *Proc. Nat. Acad. Sci.*, vol. 105, no. 41, pp. 15825–15830, 2008.
[36] S. Chevalier-Skolnikoff, "An exploration of the ontogeny of deception in human beings and nonhuman primates," in, *Deception, Perspectives on Human and Nonhuman Deceit*. Albany, NY, USA: SUNY Press, pp. 205–220, 1986.
[37] V. K. Sharma, "An easy method of constructing latin square designs balanced for the immediate residual and other order effects," *Can. J. Statist.*, vol. 3, no. 1, pp. 119–224, 1975.
[38] S. Nolfi, D. Parisi, and J. L. Elman, "Learning and evolution in neural networks," *Adaptive Behav.*, vol. 3, no. 1, pp. 5–28, 1994.
[39] S. Nolfi and D. Parisi, "Learning to adapt to changing environments in evolving neural networks," *Adaptive Behav.*, vol. 5, no. 1, pp. 75–98, 1996.
[40] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA, USA: MIT Press, 2000.
[41] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer, 1981.
[42] T. Wiltshire and S. Fiore, "Social cognitive and affective neuroscience in human-machine systems: A roadmap for improving training, human-robot interaction, and team performance," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 6, pp. 779–787, Dec. 2014.

Shir Li Wang received the B.Sc. degree in 2002 and the M.Sc. degree in 2007, both from the University of Science Malaysia, Penang, Malaysia, and the Ph.D. degree from the University of New South Wales, Canberra Campus, Canberra BC, Australia, in 2012. Her research interests include neural network, evolutionary computation, clustering, data mining, and adversarial learning.

Kamran Shafi the B.Sc. degree in electrical engineering, the M.Sc. degree in telecoms engineering, and the Ph.D. degree in computer science. He is currently a Lecturer at UNSW Australia, Canberra Campus. He is interested in the applications of computational intelligence to business analytics to provide effective solutions to real world decision problems, including national defence, logistics, and computer security. He has contributed to several disciplines including genetic-based machine learning, game theory, and optimization.

Theam Foo Ng received the B.Sc. (Hons.) mathematics, in 2002, and the M.Sc. degree in statistics, in 2006, from Universiti Sains Malaysia (USM), Penang, Malaysia. He received the Ph.D. degree from the University of New South Wales, Sydney NSW, Australia, in 2012. He is now with the Centre for Global Sustainability Studies, USM, where he is currently a Senior Lecturer. His research interests include mathematics, statistics, machine learning, pattern recognition, and education for sustainable development.

Chris Lokan received the B.Sc. and Ph.D. degrees, in 1980 and 1985, respectively, from the Australian National University, Canberra ACT, Australia. He is a Senior Lecturer at UNSW Australia, Canberra Campus. His teaching concentrates on software engineering. His main research interests include empirical software engineering, software effort and cost estimation, software benchmarking, complex adaptive systems, and data mining. He is a Member of the ACM, the IEEE Computer Society, and the Australian Software Metrics Association.

Hussein A. Abbass is a Fellow of the U.K. Operational Research Society and the Australian Computer Society. He is currently the Vice-President for Technical Activities at the IEEE Computational Intelligence Society. He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, and four other journals. His work integrates cognitive science, operations research, and artificial intelligence.