MASKS: A Multi-Artificial Neural Networks System's verification approach

Amirhoshang Hoseinpour Dehkordi,Majid Alizadeh*,Ebrahim Ardeshir-Larijani,and Ali Movaghar

***Abstract*—Artificial Neural networks are one of the most widely applied approaches for classification problems. However, developing an errorless artificial neural network is in practice impossible, due to the statistical nature of such networks. The employment of artificial neural networks in critical applications has rendered any such emerging errors, in these systems, incredibly more significant. Nevertheless, the real consequences of such errors have not been addressed, especially due to lacking verification approaches. This study aims to develop a verification method that eliminates errors through the integration of multiple artificial neural networks. In order to do this, first of all, a special property has been defined, by the authors, to extract the knowledge of these artificial neural networks. Furthermore, a multi-agent system has been designed, itself comprised of multiple artificial neural networks, in order to check whether the aforementioned special property has been satisfied, or not. Also, in order to help examine the reasoning concerning the aggregation of the distributed knowledge, itself gained through the combined effort of separate artificial neural networks and acquired external information sources, a dynamic epistemic logic-based method has been proposed. Finally, we believe aggregated knowledge may lead to self-awareness for the system. As a result, our model shall be capable of verifying specific inputs, if the cumulative knowledge of the entire system proves its correctness. In conclusion, and formulated for multi-agent systems, a knowledge-sharing algorithm (Abbr. MASKS) has been developed. Which after being applied on the MNIST dataset successfully reduced the error rate to roughly one-eighth of previous runs on individual artificial neural network in the same model.**

***Index Terms*—Artificial Neural Networks, Dynamic Epistemic Logic, Multi-Agent System, Verification.**

*Corresponding author.

Amirhoshang Hoseinpour Dehkordi was with the School of Computer Science, Institute for Research in Fundamental Sciences, Tehran, , 19395-5746 Iran e-mail: amir.hoseinpour@ipm.ir.

Majid Alizadeh was with the School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, , 14155-6455 Iran e-mail: majidalizadeh@ut.ac.ir.

Ebrahim Ardeshir-Larijani was with the School of Computer Science, Institute for Research in Fundamental Sciences, Tehran, , 19395-5746 Iran e-mail: e.a.larijani@ipm.ir.

Ali Movaghar was with the Department of Computer Engineering, Sharif University of Technology, Tehran, , 11155-9517 Iran e-mail: movaghar@sharif.edu.

I. Introduction

Artificial Neural Networks (ANNs), among other classifiers, are used in many real-world applications. Among these, and in particular, are safety-critical systems, in which failures may result in catastrophic consequences. Regarding this, ANN has concerned itself with the investigation of statistical methods that can be employed to improve the performance of certain tasks, whilst also proceeding to apply the acquired information to the decision-making process.

Usually, and concerning critical cases, errors inherent in the output of the system are compared against the output of human decisions. Yielding crucial importance regarding the further analysis of these failed cases. For example, imagine a keep-right traffic sign, which due to precipitous weather, has been mistakenly identified by an ANN as a turn-left sign, something obvious for a human. For an autonomous car's AI system, this is considered a critical error. Of course, this mistake may have a low probability of occurrence but can result in serious consequences. As pertaining to performance measures, two issues are material and may cause vulnerabilities. First, are architectural flaws and

deficiencies. And second, having limited sets for training the artificial neural networks. Nevertheless, and putting aside architectural flaws and limited data-sets, the system may even be confronted by prearranged noisy images, which have been inputted with malignant intentions (i.e. adversarial examples) [1], [2]. However, as has been observed, adversarial inputs tend to locomote themselves towards neighborhoods (which, considering predefined norms, are images located in the near vicinity). And also towards manipulations (which, considering human perception, are considered visually similar), such as camera angle changes and image skewing. So, it makes sense that any proposed property should incorporate possible neighborhoods and manipulations in order to verify the correct working of ANNs.

Here, and considering formal verification, theoretical understanding regarding the correctness of our formula gains importance. In addition, as a designer, we are inclined to identify whether a property holds true in the system or not. Ordinarily, all possible executions, in input-output based systems, are analyzed via searches conducted through the entirety of the input domain. However, in order to reach formal verification, and preceding all else, a measurable property should be defined and agreed upon. Only consequently, the verification approach can be determined.

Historical Background. Historically speaking, examples of failure in formal verification have resulted in massive financial losses. Among these failures is the Pentium bug, which surfaced around 1994 and Intel lost approximately half a billion dollars [3]. It is due to these examples that formal verification has been developed. In essence, it is a way to formally and mathematically verify the outcome of operations of systems. As these verification methods grew in popularity, so has their application on ANNs. As a basis, one can reference the safety verification on ANNs in *Multi-Layer Perceptrons Neural Networks*, a study carried out by Pulina et. al. [4], in which linear computation constraints abstracted to *Boolean*.Howbeit, their particular approach was limited, due to computational restraints, to the use of a maximum of six neurons. Advancing in time, environmental modeling, formal specification, system modeling, computational engines, and correct-by-construction design were further considered, by Sanjit et. al [5], and added into the literature. Building on the aforementioned studies, a unified framework was created, by D. Sadigh [6], in order to provide safe and reliable integration for human-robot systems. In addition, and regarding safety-critical verification of ANNs, one can address the work of Scheibler et. al. [7], in which bounded model checking has been applied. There, the formulas introduced by the verification method were solved via iSAT3 (an SMT-solver) and special deductions. Following their research, improvements were made, by Katz et. al [8], by taking into consideration more general properties. This allowed them to verify networks by using simplex methods which include piecewise linear ReLU activation functions. Their method was computationally restrained to approximately 300 neurons, which was quite an improvement. Albeit, they could not verify larger networks, such as Alexnet [9] which includes about $\sim$650,000 ReLU nodes. Pushing onward, point-wise robustness (for each layer) was introduced, by Huang et al. [10], and inserted into the verification method. Their algorithm succeeded in exhaustively searching the neighborhood of a network's inputs, with reasonable complexity and in an acceptable time-frame. The model developed by [10] appears well optimized, and more general. However, due to it not being designed for verifying safety-critical cases regarding a collaborative system of ANNs, it cannot reason about knowledge generated in such multi-agent systems.

Contributions. To overcome this problem, we developed a hybrid (ANN/Logical) solution in which the knowledge of multiple ANNs was aggregated, in order to reduce errors, in a MAS scenario. This was accomplished by defining a special property that could extract the knowledge of such ANNs. Subsequently, a multi-agent system was designed, constituted of multiple ANNs. Afterward, regarding reasoning about the aggregation of knowledge, a dynamic epistemic logic-based method was developed. This knowledge could be gained through the working of separate ANNs or acquired through external in-

formation sources. Furthermore, the aforementioned dynamic model was employed, in the context of a Kripke model, which introduced a more precise measurement concerning the knowledge of ANNs. Finally, aggregated knowledge may result in self-awareness for the system, which may occur when the model has been verified for specific inputs.

outline. In this paper, a logical approach has been assessed, to reason about knowledge of multi-ANNs, and also to establish knowledge dynamism in the system. This paper has been structured in the following manner:

First of all, a definition of ANNs has been provided (section II). Following that, the developed method, based on a multi-agent system to verify inputs by aggregation of each ANN's knowledge, has been demonstrated (section III). Next, the method was completely expressed through introduction of a collaboration algorithm (section IV). Following, we show how employment of our distributed approach could improve the results of multiple ANNs (section V). Finally, any extensions, possible future works, and the conclusion have been included (section VI).

II. Background on Artificial Neural Networks

Classification is the process of partitioning an input vector space into several classes. We notice that inputs are vectors in Euclidean n-space with a defined arbitrary distance metric. The training process in ANN, which is based on a *training* dataset, initiates with a random partitioning method. After that, the partitions update themselves according to each input point, delivered from the training dataset. Finally, the trained ANN presents the function in which the input vector space can be partitioned into a certain number of classes.

Formally, a feed-forward ANN is a tuple $N = (L, T, \Phi)$ where $L = \{L_k | k \in \{0, \cdots, n\}\}$, $n \geq 1$ is a set of layers, which contains nodes, each called neurons; $T \subseteq L \times L$ is the edge's weights of n-partite graph, and $\Phi = \{\phi_k | k \in \{0, \cdots, n\}\}$ is a set of activation functions $\phi_k : D_{L_{k-1}} \to D_{L_k}$, where $0 < k \leq n$ and $D_{L_k} \subseteq \mathbb{R}^{n_k}$ are the dimensions of k-th layer. An ANN is called a deep neural network if it has at least two layers. The size of L_k is shown by n_k. And L_0 and L_n are called the input and the output layers, respectively. For a single input x, $[x]_G$ is class label of the ANN $G \in \mathbb{G}$ which contains x. Whereas for a set of input points X, the class label is $[X]_G = \bigcup_{y \in X} [y]_G$. In this paper, and for clarification purposes, we drop the index G and write $[x]$ and $[X]$ wherever it can be understood from the context. *Hidden layers* are layers which are not input or output. The process is carried out inductively as follows: first, an input data x_{L_0} is fed into the input layer. Next, with i-th layer data, the data of $i+1$-th layer is calculated by $x_{L_{i+1}} = \phi_{i+1}(x_{L_i} \times T_{L_i, L_{i+1}})$ in which $T_{L_i, L_{i+1}}$ is the adjacency matrix of layers L_i and L_{i+1}. Then, the output class can be decided by the output layer's value (the most common way is to use maximum argument value as an output class).

III. Logic-based Model for classifiers

Typically, classification is a process in which an input vector space is partitioned into an output number of classes. Let us illustrate this with a simple example, say a face detector that takes 100 by 100 grayscale input images, which partitions a 10000 dimensions input vector space into two classes: "images with or without a face". By changing the value of each pixel of an image, the respective point moves through the input vector space (the more it changes, the farther it moves). Regarding human perception, minor changes do not alter our understanding of the image. Therefore, by defining a measure ϵ, all images located within a radius, namely the neighborhood set, shall be regarded similarly through human eyes. Furthermore, we can enrich the neighborhood set by adding are more distant in euclidean geometry, yet are still regarded as similar by humans. For instance, many objects are recognized as being the same (e.g. whether through various camera angles, at night or at day, etc), yet may be located distantly from each other, in the vector space. Now, an input point is located in the knowledge set of a classifier, exactly when all points in a neighborhood set have been classified into the same class. One consequence of this definition is that small changes do not alter the output class, if the point is located within the knowledge set of the classifier. Therefore, this method can simulate the

knowledge represented in the human brain. Figure 1 shows the knowledge obtained for input points x, y and z. In a proper classifier, most failures occur when the input point is not located in the knowledge set. Furthermore, elements in the knowledge set can be assumed as safety verified input points, as pertaining to the defined knowledge.

In addition, when all inputs of the neighborhood set do not land in the same class, these classes can be considered as alternative outputs. This dilemma can be eradicated via employing knowledge sharing between classifiers. Suppose a group of ANNs is going to share knowledge about an input point. Each ANN knows the input point's alternatives output classes. In the case that the intersect of these output classes (possible knowledge) is a single output class, such a system can be verified through applying shared knowledge. Moreover, it is shown that outsider's knowledge can also be included, in order to adjust for the system. For illustration, assume that a camera is installed to identify passing animals. For a specific input image, it may doubt whether the image is of a sparrow or bat. As an external knowledge, a zoologist informs that bats don't appear during daylight. In such a case, external knowledge can help the system in achieving correct identification of the animal.

Accordingly, our developed systems of ANNs may have miscalculation problems, if faced with the following scenarios:

1) **Wrongly verifying input points:** this occurs when an input point has been considered as robust in a wrong class (i.e. the input point and all members of the neighborhood and manipulation set lie in the wrong output class). This failure may happen when the inaccuracy of the partitioning algorithm of the ANN is more than the broadness of the neighborhood and the manipulation set. The root-cause of this can be in selecting a inadequately sized set for the neighborhood and the manipulation set, or it can result from an insufficient number of ANNs employed in the MAS.
2) **Correct answers not verified:** sometimes, input points that are correctly classified with the ANNs are not verified by the MAS. In this scenario the input point should be located near the partitioning boundaries of all the ANNs of the MAS. In other words, there exist similar inputs (which are elements of the neighborhood and manipulation set) which have been wrongly classified into the same class, by all ANNs. Through examining every intersection of output classes for the outputs of ANNs, for neighborhood and manipulation sets, a subset of classes can be collected. This subset is considered common among outputs for each ANN. Although the input point cannot be verified for a single result, it is acknowledged that the verified element resides in an subset of represented results.

The above-mentioned outline the use of the neighborhood and manipulation set as a knowledge-sharing approach and extends the definition of verification, in a MAS scenario. For this, we have developed a logic-based approach to formalize knowledge-sharing, and we've investigated a formal definition for the verification process.

Formally, the input point is in the knowledge set of a single ANN, in our proposed model, exactly when it becomes robust. To define whether an input point is robust we need to employ the notion of neighborhood. Let $X \subseteq \mathbb{R}^n$ be an input domain and $x_0 \in X$, then the *neighborhood* (ϵ- neighborhood) of x_0 is the set:

$$\eta(x_0) = \{x \in X \mid d(x_0, x) < \epsilon\},$$

where d is an arbitrary distance metric.

A point x in a given ANN G is *robust* exactly when $[\eta(x)]_G = [x]_G$.

A. Artificial Neural Network and Dynamic Epistemic Logics

In this section, we introduce a novel interpretation of dynamic epistemic logic that suits the formal description of the dynamics of the knowledge of ANNs, as the agents in a MAS. Let x be an input point and G be an ANN. We say that G verifies x exactly when x is robust in G. We also note that x is verified in a MAS $\mathbb{G}$ exactly when $\bigcap_{G \in \mathbb{G}} [x]_G$ is singleton.

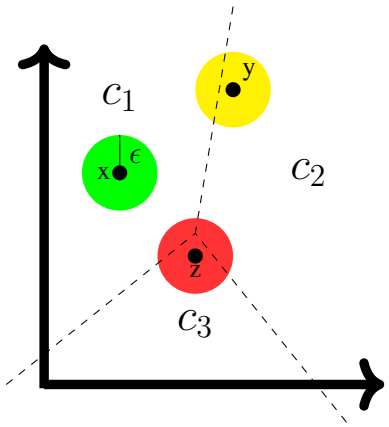

Fig. 1. This is the partitioning plot for an ANN G and 3 input points x, y, and z. Here, the input x is located robustly in class c_1, because all neighborhoods within ϵ distance are in that class, and G knows that this point is in c_1. Although y is in class c_2, there may exist some neighborhoods of y which are in class c_1. Herein, the classifier does not yet know whether the output class is c_1 or c_2, however, the classifier knows that the output class is not c_3 (i.e. $\neg(y, c_3)$, so (y, c_1) or (y, c_2) are alternatives). Meanwhile the input point z produces no knowledge in this scenario because the input's alternative outputs are matches to the total alternative output classes (i.e. $(z, c_1) \vee (z, c_2) \vee (z, c_3)$).

Here, we review the basics of Dynamic Epistemic Logic (DEL) in order to able to reason regarding the knowledge of agents. DEL is a logical framework designed to deal with the dynamics of knowledge of agents in multi-agent systems through adding dynamic modalities to epistemic logic. These modalities can be quantified over transformations of possible world models (see next subsection). Hence, the agent's actions can alter the facts of the possible worlds. For more details see [11]–[15]. The most simple version of Dynamic epistemic logic is public announcement logic, which is an extension of multi-agent epistemic logic. Where dynamic operators are employed to model the informational consequences of announcements to the entire group of agents (see [16]–[18]). In order to introduce the language of epistemic logic, we consider the set of propositional variables as a finite set of pairs of the form (x, c), made of an input point x and a class $c \in C$ in the output layer. The intended meaning of (x, c) is "at least one of the elements in $\eta(x)$, results c".

Let a set of propositional variables Φ and a finite set of agents $Ag = \{1, \ldots, n\}$ be given. The epistemic language is defined inductively through the following grammar, in BNF:

$$\Phi := \top \mid p \mid \neg\phi \mid (\phi \wedge \phi) \mid K_j\phi \mid D_A\phi,$$

where $p \in \Phi$, $j \in Ag$ and $A \subseteq ANNs$. We use the common abbreviations $\bot := \neg\top$, $\phi \vee \psi := \neg(\neg\phi \wedge \neg\psi)$, $\langle K_j\rangle\phi := \neg K_j\neg\phi$, and $E_A\phi := \bigwedge_{j\in A} K_j\phi$. The intended meaning of $k_i\phi$ is that " agent i knows ϕ, and that of $\langle K_i\rangle\phi$ is that "it is consistent for agent i to know ϕ". We now define the language of Public Announcement Logic (PAL), by adding dynamic modality, as follow:

$$\Phi ::= p \mid \neg\phi \mid (\phi \wedge \phi) \mid K_j\phi \mid D_A\phi \mid [\phi]\phi.$$

An inference system for PAL can be found in [19].

Moving on to the semantics of such a logic, models for epistemic logic are tuples $\mathcal{M} = (W, R_1, \cdots, R_n, V)$, known as Kripke models, where $W = \{w_0, w_1, \cdots, w_k\}$ is a set of worlds or states (in our study, W, represents all possible output results for the input values), $R_i \subseteq W \times W$ is the equivalence relation for every agent i and $V : W \to 2^\Phi$ is the evaluation function. The fact that sR_is' is taken to mean that the agent i cannot tell states s and s' apart. Before addressing the formal definition of satisfaction in a model, it is noteworthy that for each input point x_0, the possible worlds shall be representing all epistemic possible states (i.e. they show all possible subsets of output class set). These possible worlds can be filtered through the neighborhood and manipulation set $\eta(x_0)$. Let C be the set of all output classes. If $|\eta(x_0)| \geq |C|$, the number of all possible worlds shall be $2^{|C|}$ (generally $|\eta(x_0)| \gg |C|$ can be assumed). Therefore, the satisfaction of an atomic formula (x_0, c_i) in a state w means that world c_i has appeared as an output class of the ANN G_k, for some point of $\eta(x_0)$. For instance, if c_i, c_j and c_k appear in the output classes of G_k, for all members of $\eta(x_0)$, then the *actual world* [15] can satisfy (x_0, c_i), (x_0, c_j) and (x_0, c_k).

After which, we attempt to formally define what it means for a formula ϕ to be true in $w \in W$, written $\mathcal{M}, w \models \phi$, inductively as follows:

- $\mathcal{M}, w \models p$ iff $p \in V(w)$,
- $\mathcal{M}, w \models \neg\phi$ iff $\mathcal{M}, w \not\models \varphi$,
- $\mathcal{M}, w \models \phi \wedge \psi$ iff $\mathcal{M}, w \models \phi$ and $\mathcal{M}, w \models \psi$,
- $\mathcal{M}, w \models K_j\phi$ iff $\forall v \in R_j(w), \mathcal{M}, v \models \phi$,
- $\mathcal{M}, w \models D_A\phi$ iff $\forall v \in R_{D_A}(w), \mathcal{M}, v \models \phi$, where $R_{D_A} := \bigcap_{i \in A} R_i$.
- $\mathcal{M}, w \models [\psi]\phi$ iff $\mathcal{M}, w \models \psi$ implies $\mathcal{M}^\psi, w \models \phi$.

where $\mathcal{M}^\psi$ is the updated Kripke model $\mathcal{M}$ by the announcement ψ with $W^{\mathcal{M}^\psi} := \{w \in W \mid (\mathcal{M}, w) \models \psi\}$, the relation $R_i^{\mathcal{M}^\psi} := R_i \cap (W^{\mathcal{M}^\psi} \times W^{\mathcal{M}^\psi})$ and valuation $V^{\mathcal{M}^\psi}$ which is the the valuation V restricted to $W^{\mathcal{M}^\psi}$ [20].

Here, we concentrate on the interplay of knowledge and epistemic action, which only modifies the agents' knowledge while leaving the facts unchanged. So, $[\psi]$ is an operator that takes us to a new model consisting only of those worlds where ψ has been rendered as true. Therefore, after the announcement of ψ, no agent considers worlds where ψ was false. Hence, we should evaluate formulas in the sub-model $\mathcal{M}^\psi$. We also notice that the operator D_A can be interpreted as a necessity operator of the relation on R_{D_A}.

If we consider all such Kripke models, the set of all valid formulas obtained from these semantics is known as modal logic $\mathbf{S5}$, see [21].

B. State Space Reduction

In the model above, the cardinality of the model grows exponentially, as pertaining to the number of output classes. This could result in space state explosion, especially for large number of output classes. For a given set of output classes C, the state space of the model is constructed by the set of all subset of C, i.e., the power-set $\mathcal{P}(C)$. However, $\mathcal{P}(C)$ is completely characterized by its singleton elements. Therefore we can ignore any element of the model, other than the singleton ones. Hence we obtain a reduced model with a cardinality of $|C|$, with relations and valuation restricted to C. Noticing that since the accessibility relations in the original model are equivalence relations, we can retrieve all the information of the original models (the relations and the valuation function) through the reduced model. We formalize this idea in the following way, let the model $\mathcal{M} = (W, R_1, \cdots, R_n, V)$ be given, for $w_i \in W$ we define $V_\Phi(w_i) = \{(x, c) \in \Phi \mid \mathcal{M}, w_i \models (x, c)\}$. Then the reduced model M^r is defined as:

- $W^r := \{w_i \in W \mid |V_\Phi(w_i)| = 1\}$,
- $R_i^r := R_i \cap (W^r \times W^r)$, for any agent i, and
- $V_\Phi^r(w) = V_\Phi(w)$, for every $w \in W^r$.

Now we can retrieve the state $w_j \in W \setminus W^r$ as follows, let $V_\Phi(w_j) = \{(x, c_{j_1}), (x, c_{j_2}), \cdots, (x, c_{j_k})\}$. Then w_j is completely characterized by $w_{j_1}, w_{j_2}, \cdots, w_{j_k} \in W^r$, where $V(w_{j_t}) = \{(x, c_{j_t})\}$. Next, if $w_{j'} \in W \setminus W^r$, with $V_\Phi(w_{j'}) = \{(x, c_{j'_1}), (x, c_{j'_2}), \cdots, (x, c_{j'_{k'}})\}$ and characterized by $w_{j'_1}, w_{j'_2}, \cdots, w_{j'_{k'}} \in W^r$, then $w_j R_i w_{j'}$, for an agent i, exactly when $w_{j_1} R_i^r w_{j_2} R_i^r \cdots R_i^r w_{j_k} R_i^r w_{j'_1} R_i^r w_{j'_2} R_i^r \cdots R_i^r w_{j'_{k'}}$ in M^r.

In the following theorem, we state the relationship of the satisfaction problem between the original model $\mathcal{M}$ and the reduced model $\mathcal{M}^r$.

Theorem III.1. *Let $\mathcal{M} = (W, R_1, \cdots, R_n, V)$ be a Kripke model. Then for any $w \in W$ and any formula θ with $V_\Phi(w) = \{(x, c_1), (x, c_2), \cdots, (x, c_t)\}$ we have $\mathcal{M}, w \vDash \theta$ exactly when $\mathcal{M}^r, w_i \vDash \theta$, where $V_\Phi(w_i) = \{(x, c_i)\}, 1 \leq i \leq t$.*

Proof. We complete the proof, by induction on the complexity of θ that for all $w \in W$ with $V_\Phi(w) = \{(x, c_1), (x, c_2), \cdots, (x, c_t)\}$ we have $\mathcal{M}, w \vDash \theta$ exactly when, $\mathcal{M}^r, w_i \vDash \theta$. We only consider the interesting case $\theta = [\psi]\varphi$, and other cases are almost trivial.

If direction: suppose that $\mathcal{M}, w \vDash [\psi]\varphi$, if $\mathcal{M}, w \vDash \psi$ then by induction we have $\mathcal{M}^r, w_i \vDash \psi$ for all $1 \leq i \leq t$, which implies that, by induction again, $(\mathcal{M}^r)^\psi, w_i \vDash \varphi$, since $\mathcal{M}^\psi, w \vDash \varphi$. If not, there exists $1 \leq i \leq t$ such that $\mathcal{M}^r, w_i \nvDash \psi$ then by definition $\mathcal{M}^r, w_i \vDash [\psi]\varphi$. If $\mathcal{M}, w \nvDash \psi$, then there exists $S \subseteq \{1, \cdots, t\}$ such that for all $i \in S$, $\mathcal{M}^r, w_i \nvDash \psi$. Thus, we have $\mathcal{M}^r, w_i \vDash [\psi]\phi$, for $i \in S$. Now, for $i \in S' = \{1, \cdots, t\} \setminus S$ we have $\mathcal{M}^r, w_i \vDash \psi$. Then, by induction, we have

$\mathcal{M}, w_i \vDash \psi$, for $i \in S'$. Now, since $\mathcal{M}^r, w_i \vDash \psi$, $i \in S'$ then there exists $w' \in W$ such that $V_\Phi(w') = \{(x, c_i) \mid i \in S'\}$. But, by assumption, we have $\mathcal{M}, w' \vDash [\psi]\varphi$ which implies that $\mathcal{M}^\psi, w' \vDash \varphi$. Then, for all $i \in S'$, $(\mathcal{M}^r)^\psi, w_i \vDash \varphi$ and therefore $(\mathcal{M}^r)^\psi, w_i \vDash [\psi]\varphi$.

Only if direction: suppose that $\mathcal{M}^r, w_i \vDash [\psi]\varphi$ for all $1 \leq i \leq t$. Let $S = \{j \mid \mathcal{M}^r, w_j \vDash \psi\}$, and $S' = \{1, \cdots, t\} \setminus S$. Then, by the assumption, $(\mathcal{M}^r)^\psi, w_j \vDash \varphi$, for $j \in S$ which implies, by induction, that $\mathcal{M}, w \vDash \psi$ and $\mathcal{M}^\psi, w \vDash \varphi$. Therefore, $\mathcal{M}^\psi, w \vDash [\psi]\varphi$ for all w with $V_\Phi(w) \subseteq \{(x, c_j) \mid j \in S\}$. If $\mathcal{M}^r, w_{j'} \nvDash \psi$, then for all $\{w \mid (x, c_{j'}) \in V_\Phi(w),\ j' \in S'\}$ we have $\mathcal{M}, w \nvDash \psi$. Hence, $\mathcal{M}, w \nvDash [\psi]\varphi$. □

C. 2-Multi-ANN Systems

In the DEL, the interpretation of external knowledge is also considered feasible. This means that in a MAS, besides the internal distributed knowledge, outer knowledge sharing can be investigated, in order to reduce the possible worlds. For example, in an optical character recognition (OCR) problem, limited to numbers, knowledge of the existence of a circle in the shape of the input image could reduce to four possible worlds (0,6,8 and 9, which have at least a loop in their shape). Another benefit of considering external knowledge is the ability to divide a problem into smaller parts and also to solve more simplistic problems using various MAS and the sharing of knowledge to conquer the problem. For this kind of divide-and-conquer algorithm in the MAS scenario, first, the problem should be broken down into simpler parts and the rule of division should be collected (here, rules are the restriction applied to the process of combination of the divisions), where each division can be solved with a MAS of ANNs. Second, all MASs should publicly announce their remaining possible worlds as external knowledge. Next, considering the rules, impossible combinations should be avoided to reduce the number of possible worlds. Finally, when one and only one possible world exists, it can be verified deterministically. For example, assume two classifiers that are supposed to classify an input image, one of which classifies the background, and another that does the same for the foreground. Next, suppose that the first classifier has identified the background as a city street, and the second is uncertain, regarding the foreground, between the existence of an elephant or a car. Using external knowledge that implies, "elephants do not wander in city-streets", the only possible world for the foreground would be: there is a *car* in the street.

Formally, Assume that

$$\mathcal{M}_k = (W_k, R_{k,1}, \cdots R_{k,n_k}, V_k),$$

where $1 \leq k \leq n$, are given. The Kripke model:

$$\mathcal{M} = (W, R_1, \cdots, R_n, V)$$

that models knowledge-sharing between ANNs, is defined through the following components:

- $W = W_1 \times \cdots \times W_n$,
- $(w_{i_1}, \cdots, w_{i_n}) R_k (w_{j_1}, \cdots, w_{j_n})$, for $1 \leq k \leq n$, exactly when for all $l \neq k$ we have $w_{i_l} = w_{j_l}$ and there exists $1 \leq m \leq n_k$ such that $w_{i_k} R_{k,m} w_{j_k}$,
- $V(w_1, \cdots, w_n) = (V_1(w_1), \cdots, V_n(w_n))$.

IV. Artificial Neural Networks Collaboration

Suppose that a group of trusted ANNs work together towards achieving a more self-aware system (where trusted ANNs share the entire owned knowledge correctly). To do so, first of all, in the algorithm 1 the process of knowledge extraction from one agent-input is developed, i.e., we collect all output classes of inputs in $\eta(x)$ related to the given ANN. In this algorithm, an ANN is a function $\mathcal{N}(x)$ for input x, and the output result of the function represents the output class of x. Consequently, $\mathcal{K}$ represents the knowledge of $\mathcal{N}$ with the above-mentioned definitions. As a result, the output represents whether the investigated ANN is robust, for the input x, or not. And it includes the possible answers of the set $\eta(x)$ from the agent's perspective.

After obtaining the knowledge of each agent (i.e., ANN) with algorithm 1, the algorithm 2 has been developed to aggregate all the knowledge of these agents. This knowledge also demonstrates the possible worlds, from the perspective of each agent. Herein, by determining the intersected knowledge of all ANNs in the MAS, the result of verification, and

Algorithm 1 The ANN Knowledge Calculator (NNKC) function shall calculate the knowledge, produced by an ANN, for an input point, using a neighborhood function

Let $\mathcal{N}(x) = c$ be the function of the considered ANN and c be the result class

function NNKC($\mathcal{N}, x_0, \eta$)
    ▷ $\mathcal{N}, x_0, \eta$ are ANNs, input point and neighborhood function respectively
    $\mathcal{K} \leftarrow \emptyset$
    for all $x \in \eta(x_0)$ **do**
        $c \leftarrow N(x)$ ▷ c represents the respective possible world
        if $c \notin \mathcal{K}$ **then**
            Add c to $\mathcal{K}$ set
    if $|\mathcal{K}| = 1$ **then**
        return 1, $\mathcal{K}$
    return 0, $\mathcal{K}$

the aggregated knowledge from the group of agents can be measured. As an output, if the intersection results in one class, the input finds itself verified. Otherwise, if more than one class exists in the intersection result, the input point cannot be verified. Although, the set of possible classes represents the possible verified outputs for the MAS. Finally, if an empty set returns as an output, inconsistency emerges in the MAS's agent knowledge, and input, for that particular case, cannot be verified.

In the algorithm 3, $\mathcal{K}_\mathcal{S}$ shows the remaining possible worlds in which verification formulas can be satisfied.

V. Two Examples

Reducing the error rate of ANNs is crucial in critical scenarios. In our method, a MAS of ANNs has been offered for this exact purpose. In the sequel, we will give two examples explaining the details of our proposed model. In the first, we demonstrate exactly how the knowledge set, in the context of dynamic epistemic logic, can be defined. In the second example, however, we demonstrate the usability of our model in real-world practical applications, i.e., a self-aware MAS is developed.

Algorithm 2 The MAS Knowledge Aggregator (MASKA) function is going to aggregate the knowledge that is produced by a MAS of ANNs, for each input point using a neighborhood function

function MASKA($\mathcal{N}_\mathcal{S}, x_0, \eta$)
 ▷ $\mathcal{N}_\mathcal{S}, x_0, \eta$ are groups of ANNs, input points and neighborhood functions respectively
    $\mathcal{K}_\mathcal{S} \leftarrow \emptyset$
    for all $\mathcal{N} \in \mathcal{N}_\mathcal{S}$ **do**
        is_Robust, $\mathcal{K} \leftarrow$NNKC($\mathcal{N}, x_0, \eta$)
        $\mathcal{K}_\mathcal{S} \leftarrow \mathcal{K}_\mathcal{S} \cap \mathcal{K}$
        if $\mathcal{K}_\mathcal{S} = \emptyset$ **then**
            return 0, $\emptyset$
    if $|\mathcal{K}_\mathcal{S}| = 1$ **then**
        return 1, $\mathcal{K}_\mathcal{S}$
    return 0, $\mathcal{K}_\mathcal{S}$

Algorithm 3 The MAS Knowledge Sharing (MASKS) function shall aggregate the knowledge that is produced by an external system

function MASKS($\mathcal{N}_\mathcal{S}, x_0, \eta$)
 ▷ $\mathcal{N}_\mathcal{S}, x_0, \eta$ are group of ANNs, input points and neighborhood functions respectively
    $\mathcal{K}_\mathcal{S} \leftarrow \emptyset$
    for all $\mathcal{N} \in \mathcal{N}_\mathcal{S}$ **do**
        is_Robust, $\mathcal{K} \leftarrow$NNKC($\mathcal{N}, x_0, \eta$)
        $\mathcal{K}_\mathcal{S} \leftarrow \mathcal{K}_\mathcal{S} \cap \mathcal{K}$
        if $\mathcal{K}_\mathcal{S} = \emptyset$ **then**
            return 0, $\emptyset$
    if $|\mathcal{K}_\mathcal{S}| = 1$ **then**
▷ Check whether the knowledge can verify the input, or if more knowledge is needed
        return 1, $\mathcal{K}_\mathcal{S}$
    for all $\mathcal{M} \in$ All knowledge sources **do**
        is_Robust, $\mathcal{K} \leftarrow$ Announced knowledge
 ▷ The external knowledge must be written in DEL formula,
▷ where possible worlds are ones in which the formula is satisfied.
        $\mathcal{K}_\mathcal{S} \leftarrow \mathcal{K}_\mathcal{S} \cap \mathcal{K}$
        if $\mathcal{K}_\mathcal{S} = \emptyset$ **then**
            return 0, $\emptyset$
    if $|\mathcal{K}_\mathcal{S}| = 1$ **then**
        return 1, $\mathcal{K}_\mathcal{S}$
    return 0, $\mathcal{K}_\mathcal{S}$

Example V.1. *(Digit Recognition)*

In this example, we are going to develop a scenario to clarify the approach, mentioned in previous chapters. Herein, the input will lie in the domain of images, which are digits ranging from 0 to 9. So the cardinality of our Kripke model is 2^{10} and its reduced model has only 10 states $W = \{w_0, \cdots, w_9\}$ in which each world represents the existence of one digit, where $V_\Phi(w_i) = \{(x, i)\}$. For a fixed input of x, we denote the atomic formula (x, i) by i. Let the figure of the digit "0" be an input image, with three agents A_0, A_1 and A_2 as ANNs in the multi-agent system A given. A_0 produced knowledge in which "0", "6", "8" and "9" are possible answers. Figure 2 depicts the possible worlds and relations of the model for A_0. The same input for A_1, results "0", "2", "4", "6", and "8" as possible answers. The intersection of these two agents' possible answers will be w_0, w_6 and w_8 3. The last agent, A_2, ignores the world w_6 and w_8 as possible results. So, the model verifies the answer that the input case is "0". If the last agent, A_2 just rejects w_8, the input fails to be verified by the model.

Now, assume that the image is a part of a two-digit number and another digit has been classified by another process (with the multi-ANN system B). In this process, the digits "0" and "3" are recognized as possible outputs. The model for these two MAS is depicted in 4. Now, suppose an external knowledge announces that there resides, at least, one zero in the digits. By announcing this, as a fact, the world w_{36} fails to be possible. Hence we have the updated model, see fig 5). Next, we are going to reason about the answer via employing more complex external knowledge. Suppose that, the external knowledge is "none of the A and B's certainty could lead the whole system into a verified answer". In this model, we know that if B gains certainty about 3 the verified answer will be 30 (i.e. $[K_B 3]30$). Similarly, if A gains certainty about 6 the verified answer will be 06 (i.e. $[K_A 6]06$). By aggregation of these two formulae, A's certainty about 6 and B's about 3, the whole system is routed towards a verified answer. The remaining possible (i.e., verified) number is 00, in which no one's certainty can result in a single answer (i.e. $[K_A 0](06 \vee 00)$ and $[K_B 0](30 \vee 00)$).

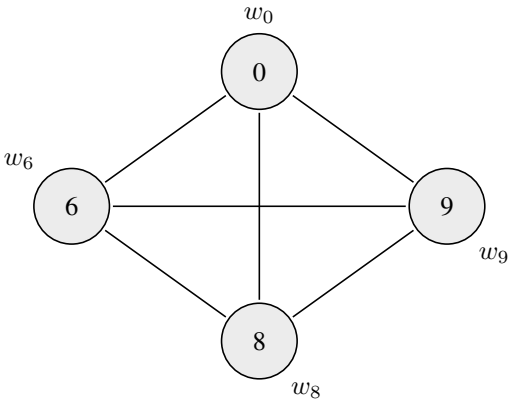

Fig. 2. The Epistemic model, considering the ANN A_0's knowledge.
$W^{(0\vee 6\vee 8\vee 9)} = \{w_0, w_6, w_8, w_9\}$;
$R_{A_0} = \{(0,6), (0,8), (0,9), (6,8), (6,9), (8,9)\}$;

Example V.2. *(Safety Verification of MNIST classification using collaboration)*

In this example, a MAS of ANNs has been developed to classify the MNIST dataset into 10 distinct classes. The system contains one to 680 randomly generated ANNs (see V.3) each with $\sim$ 98% accuracy. Here, each input and any neighborhood sets (which is an affine transformation with an address of 5) are classified by individual ANNs. The result of our employed method has been depicted in Fig 6, 7 and 8. As it is shown in Fig 6, for a system with one classifier, the number of wrong robust input cases is 38. By increasing the number of agents, eventually, by employing 199 ANNs the error cases can be reduced to 32. Although the number of truth values may also decrease, the trend is not as steep as the slope of the false rate. The reduction trend of error, by truth value, has been depicted in Fig 8. Afterwards, when we developed a MAS with 680 agents, the error cases decreased to 20. Our results demonstrate that: the trend of error decrements with an increase in the number of agents.

Remark V.3. *MNIST is a standard handwriting dataset, in which 50,000 cases were used for training models and 10,000 for testing [22]. All generated*

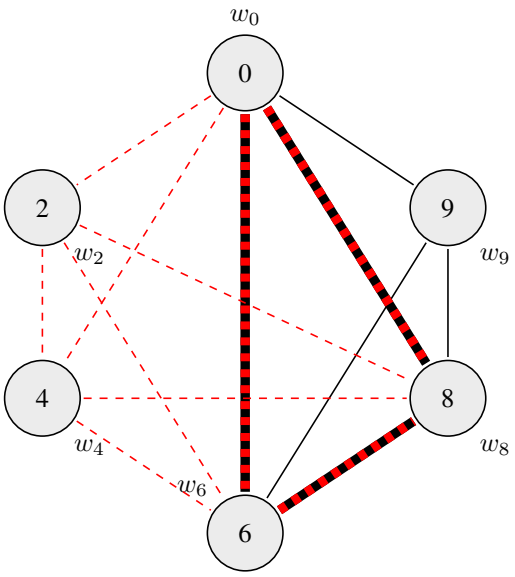

Fig. 3. The updated model, after A_1's announcement. The red-black relations are intersected relations of the model, the possible worlds after the announcement will be w_0, w_6, and w_8
$W^{(0\vee6\vee8\vee9)\wedge(0\vee2\vee4\vee6\vee8)} = \{w_0, w_6, w_8\}$;
$R_{A_0} \cap R_{A_1} = \{(0,6),(0,8),(6,8)\}$;

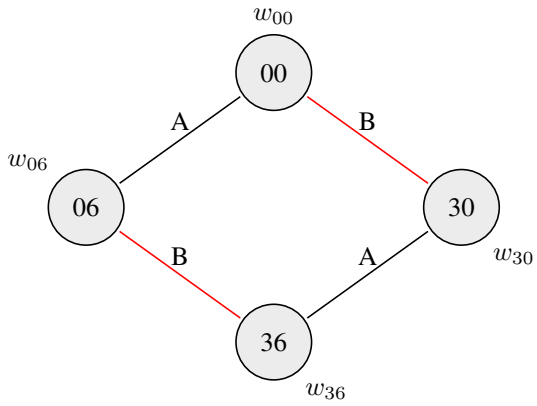

Fig. 4. The Epistemic model, a combination of MAS A and B.
$W = \{w_{00}, w_{06}, w_{30}, w_{36}\}$;
$R_A = \{(00,06),(30,36)\}$;
$R_B = \{(00,30),(06,36)\}$;
$V(w_{00})$ =00, $V(w_{06})$ =06, $V(w_{36})$ =36, $V(w_{30})$ =30;

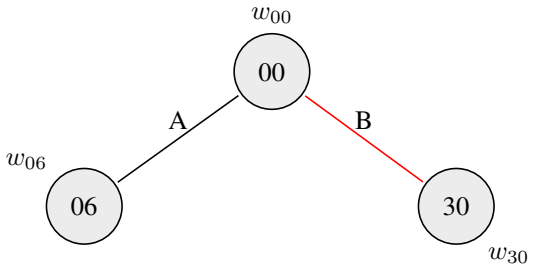

Fig. 5. The Epistemic model, after elimination of w_{36}.
$W = \{w_{00}, w_{06}, w_{30}\}$;
$R_A = \{(00,06)\}$;
$R_B = \{(00,30)\}$;
$V(w_{00})$ =00, $V(w_{06})$ =06, $V(w_{30})$ =30;

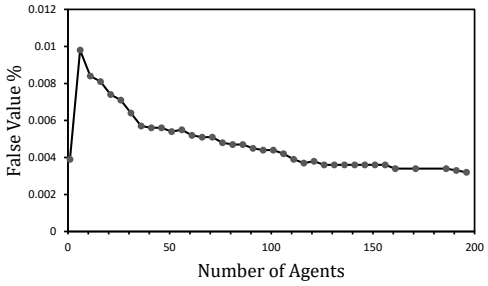

Fig. 6. Error rate percentage for various number of ANNs

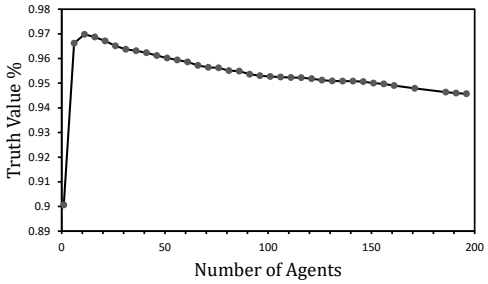

Fig. 7. Truth rate percentage for various number of ANNs

ANN models have been created, randomly, to classify the MNIST dataset using the Keras library with tensorflow as a backend [23]. The generated ANNs include 6 to 7 layers, each with 64 to 512 neurons in each layer, with all parameters being settled using the uniform random function. Subsequently,

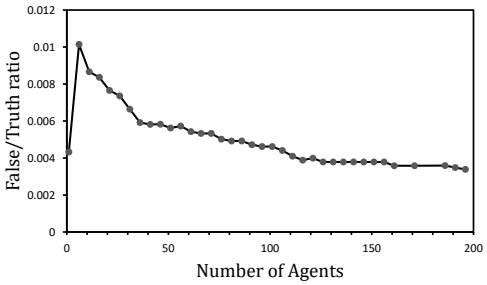

Fig. 8. Error:Truth rate percentage for various number of ANNs

only ANNs with ∼98% accuracy were chosen. The neighborhood set in V.2 has been generated by an affine transform with 5 steps, from -0.2 to 0.2. All computations in V.2 have been implemented using a computation node with the following hardware, CPU:Xeon2697V3, GPU:Nvidia GTX 1080 Ti and Ram:128GB.

VI. Conclusion and Future Works

This study aims to develop a verification method to eliminate errors through integrating multiple artificial neural networks. To do this, first, a property has been defined to present the knowledge of the artificial neural networks. Next, a multi-agent system was designed in order to investigate these multiple artificial neural networks. Then, a dynamic epistemic logic-based method was developed for reasoning about the aggregation of distributed knowledge. This knowledge, was both acquired through separate artificial neural networks, and also through external information sources. Finally, it has been shown that aggregated knowledge may lead to self-awareness for the system. As a result, the model could verify a specific input, if the knowledge of the entire system satisfies its correctness. To conclude, a multi-agent system for the knowledge sharing (MASKS) algorithm has been proposed for the aforementioned model. This proposed method was applied to the MNIST dataset, as a result, the error rate of the entire system dropped from 2% to about 0.2%.

In future, we aim to develop an approach that can model timed-series classifiers (i.e., for *recurrent neural networks* or *reinforcement learning*) for real-time verifying approaches. Meanwhile, however, a tool will be developed to verify the inputs of any kind of multi-agent system and, furthermore, to check whether an input point can be verified in the system, or not. This tool should be able to manipulate knowledge sharing, in trusted or untrusted networks. To enhance the performance of this tool, fuzzy logic must be applied to avoid state space exposure for ANNs, especially where more than two output classes exist. Similarly, a large neighborhood set may cause high loads, when applied to real-world problems. Consequently, we should define robustness, for ANNs, whilst considering pre-defined confidence levels. To do this, we must take into consideration the subset of the neighborhood, in our verification method. This may very well pave the way for defining an approximation approach for our verification methodology.

References

[1] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[3] Alexander Wolfe. Intel fixes a pentium fpu glitch. *Electronic Engineering Times*, (822):1–2, 1994.

[4] Luca Pulina and Armando Tacchella. An abstraction-refinement approach to verification of artificial neural networks. In Tayssir Touili, Byron Cook, and Paul B. Jackson, editors, *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, volume 6174 of *Lecture Notes in Computer Science*, pages 243–257. Springer, 2010.

[5] Sanjit A. Seshia and Dorsa Sadigh. Towards verified artificial intelligence. *CoRR*, abs/1606.08514, 2016.

[6] Dorsa Sadigh. *Safe and Interactive Autonomy: Control, Learning, and Verification*. PhD thesis, UC Berkeley, 2017.

[7] Karsten Scheibler, Leonore Winterer, Ralf Wimmer, and Bernd Becker. Towards verification of artificial neural networks. In Ulrich Heinkel, Daniel Kriesten, and Marko Rößler, editors, *Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, MBMV 2015, Chemnitz, Germany, March 3-4, 2015.*, pages 30–40. Sächsische Landesbibliothek, 2015.

[8] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In Majumdar and Kuncak [24], pages 97–117.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In Majumdar and Kuncak [24], pages 3–29.

[11] Johan Van Benthem, Jan Van Eijck, and Barteld Kooi. Logics of communication and change. *Information and computation*, 204(11):1620–1662, 2006.

[12] PHILIPPE BALBIANI, ALEXANDRU BALTAG, HANS VAN DITMARSCH, ANDREAS HERZIG, TOMOHIRO HOSHI, and TIAGO DE LIMA. 'knowable' as 'known after an announcement'. *The Review of Symbolic Logic*, 1(3):305–334, 2008.

[13] Rasmus Rendsvig and John Symons. Epistemic logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2019 edition, 2019.

[14] Alexandru Baltag and Bryan Renne. Dynamic epistemic logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edition, 2016.

[15] Christopher Menzel. Possible worlds. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edition, 2017.

[16] Jan Plaza. Logics of public announcements. In *Proceedings 4th International Symposium on Methodologies for Intelligent Systems*, 1989.

[17] Jan Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.

[18] Hans P Van Ditmarsch. Comments to 'logics of public communications'. *Synthese*, 158(2):181–187, 2007.

[19] Yanjing Wang and Qinxiang Cao. On axiomatizations of public announcement logic. *Synthese*, 190(1):103–134, 2013.

[20] Yí N Wáng and Thomas Ågotnes. Public announcement logic with distributed knowledge. In *International Workshop on Logic, Rationality and Interaction*, pages 328–341. Springer, 2011.

[21] Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.

[22] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[23] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[24] Rupak Majumdar and Viktor Kuncak, editors. *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I*, volume 10426 of *Lecture Notes in Computer Science*. Springer, 2017.