Possibilities and Challenges for Artificial Intelligence in Military Applications

Dr Peter Svenmarck, Dr Linus Luotsinen, Dr Mattias Nilsson, Dr Johan Schubert
Swedish Defence Research Agency SE-164 90 Stockholm
SWEDEN

{peter.svenmarck, linus.luotsinen, mattias.nilsson, johan.schubert}@foi.se

ABSTRACT

Recent developments in artificial intelligence (AI) have resulted in a breakthrough for many classical AI-applications, such as computer vision, natural language processing, robotics, and data mining. Therefore, there are many efforts to exploit these developments for military applications, such as surveillance, reconnaissance, threat evaluation, underwater mine warfare, cyber security, intelligence analysis, command and control, and education and training. However, despite the possibilities for AI in military applications, there are many challenges to consider. For instance, 1) high risks means that military AI-systems need to be transparent to gain decision maker trust and to facilitate risk analysis; this is a challenge since many AI-techniques are black boxes that lack sufficient transparency, 2) military AI-systems need to be robust and reliable; this is a challenge since it has been shown that AI-techniques may be vulnerable to imperceptible manipulations of input data even without any knowledge about the AI-technique that is used, and 3) many AI-techniques are based on machine learning that requires large amounts of training data; this is challenge since there is often a lack of sufficient data in military applications. This paper present results from ongoing projects to identity possibilities for AI in military applications, as well as how to address these challenges.

1 INTRODUCTION

Artificial intelligence (AI), specifically the subfields machine learning (ML) and deep learning (DL), has within a decade moved from prototyping at research institutes and universities to industry and real-world application. Modern AI using DL-techniques has revolutionized the performance of traditional AI-applications such as machine translation [10], QA-systems [62], and speech recognition [1]. The many advancements in this field has also turned other ingenious ideas into remarkable AI-applications capable of image caption- ing [61], lip reading [2], voice imitation [52], video synthesis [57], continuous control [7], etc. These results suggest that a machine capable of programming itself has the potential to: 1) improve efficiency with respect the development costs of both software and hardware, 2) perform specific tasks at a superhuman level, 3) provide creative solutions to problems not previously considered by humans, and 4) provide objective and fair decisions where humans are known for being subjective, biased, unfair, corrupt, etc.

In a military context, the potential for AI is present in all domains (i.e. land, sea, air, space and informa- tion) and all levels of warfare (i.e. political, strategic, operational and tactical). For instance, at the political and strategical levels, AI can be used to destabilize an opponent by producing and publishing massive quan- tities of fake information. In this case, AI will most likely also be the best candidate to defend against such attacks. At the tactical level, AI can improve partly autonomous control in unmanned systems so that human operators can operate unmanned systems more efficiently to, ultimately, increase battlefield impact.

However, as we will point out in this work, there are several key challenges that could potentially slow down or otherwise limit the use of modern AI in military applications

- Insufficient transparency and interpretability of ML-models. As an example, using DL to model the control of a self-driving car using a deep neural network (DNN) requires several hundreds of thousands of parameters [7]. Clearly such a complex program can not easily be interpreted. Even models generated using alternative ML-algorithms where the model can be graphically visualized, such as parser trees or decision trees, are hard if not impossible to interpret even when applied to toy-problems [35]. A related and perhaps even more important challenge is the ability, or in this case inability, for the AI-system to explain its reasoning to the decision maker or human operator.
- Models developed using ML are known to be vulnerable to adversarial attacks. For instance, a DL-based model can easily be deceived through manipulation of the input signal even if the model is unknown to the attacker. As an example, unmanned aerial vehicles (UAVs) using state-of-the-art object detection can potentially be deceived by a carefully designed camouflage pattern on the ground.
- The main ingredient in any ML-application is data from which the machines can learn and, ultimately, provide insight into. Military organizations are often good at collecting data for debriefing or reconstruction purposes. However, there is no guarantee that the same data can be used successfully for ML. As a result, military organizations may have to adapt their data collection processes to take full advantage of modern AI-techniques, such as DL.

The purpose of this paper is to highlight possibilities and major challenges for AI in military appli- cations. Section 2 provides a brief introduction to DL, which is the main AI-technique of interest in this paper. Section 3 provides a few examples of military AI-applications. Section 4 describes key challenges associated with AI in the military domain, as well as techniques that can be used to partially address these challenges. Conclusions are presented in Section 5.

2 DEEP LEARNING

By DL we refer to machine learning models consisting of multiple of layers of nonlinear processing units. Typically, these models are represented by artificial neural networks. In this context, a neuron refers to a single computation unit where the output is a weighted sum of inputs that passed a (nonlinear) activation function (e.g., a function that passes the signal only if it is positive). DNNs refer to systems with a large number of serially connected layers of parallel-connected neurons. The contrast to a DNN is a shallow neural network that has only one layer of parallel-connected neurons.

Until about ten years ago, training of DNNs was virtually impossible. The first successful training strategies for deep networks were based on training one layer at a time [21, 6]. The parameters of the layer- by-layer-trained deep networks were finally fine-tuned (simultaneously) using stochastic gradient methods [49] to maximize the classification accuracy. Since then, many research advances have made it possible to directly train DNNs without having a layer-by-layer training. For example, it has been found that initializa- tion strategies for the weights of the network in combination with activation function selection are crucial [16]. Even techniques such as randomly disabling neurons during the training phase [22], and normalizing the signals before they reach the activation functions [25] have shown to be of great importance in achieving good results with DNNs.

Representation learning is one of main reasons for the high performance of DNNs. Using DL and DNNs it is no longer necessary to manually craft the features required to learn a specific task. Instead, discriminating features are automatically learned during the training of a DNN.

Techniques and tools supporting DL-applications are more available today than ever before. Advanced DL can be successfully applied and customized using only limited programming/scripting skills through cheap computational resources, free ML-frameworks, pre-trained models, open-source data and code.

3 MILITARY AI-APPLICATIONS

This section presents a few examples where AI can be applied to enhance military capability.

3.1 Surveillance

Maritime surveillance is performed using fixed radar stations, patrol aircrafts, ships, and in recent years electronic tracking for maritime vessels using the automatic identification system (AIS). These information sources provide voluminous amounts of information about vessel movement that may reveal illegal, unsafe, threatening, and anomalous behavior. However, the large amounts of information about vessel movement makes it difficult to manually detect such behavior. Instead ML-approaches are used to generate normality models from vessel movement data. Any vessel movement that deviates from the normality models is considered anomalous and presented to operators for manual inspection.

An early approach to maritime anomaly detection use the Fuzzy ARTMAP neural network architecture to model normal vessel speed based on port location [47]. Another approach use associative learning of motion patterns to predict vessel movement based on its current location and direction of travel [48]. Others use unsupervised clustering based on Gaussian mixture models (GMM) [30] and kernel density estimation (KDE) [31]. The models enable detection of vessels that change direction, cross sea lanes, move in the opposite direction or travel at high speed. More recent approaches use Bayesian networks to detect false ship type, as well as discontinuous, impossible, and loitering vessel movement [36]. Future developments of maritime anomaly detection should also consider surrounding vessels and interaction among multiple vessels.

3.2 Underwater mine warfare

Underwater mines pose a significant threat to marine vessels and are used to channel movement or deny passage through restricted waters. Mine countermeasures (MCM) therefore tries to locate and neutralize mines to enable freedom of movement. Mine searches are increasingly performed with an autonomous underwater vehicle (AUV) that is equipped with synthetic aperture sonar (SAS), which provides centimeter-resolution acoustic imagery of the seafloor. Since AUVs collect large amounts of SAS imagery, automatic target classification is useful to discriminate potential mines from other objects. While automatic target classification of mines has been studied for a long time, the high performance of DNNs for image classification has created an interest in how such approaches may be useful for automatic mine detection.

A few studies show the potential of DNN for mine detection. For example, [63] describes how dummy mine shapes, mine-like targets, man-made objects and rocks where placed on the seafloor on various geo- graphic locations. An AUV was then used to survey the seafloor with an SAS. The results show that the DNN has significantly higher performance with higher probability of detection of mine shapes and lower false alarm rates compared to a traditional target classifier. Similarly, [12] describes how to generate syn- thetic SAS images of cylinder-shaped objects and various seafloor landscapes that were used to train the DNN. Further studies may investigate how to discriminate mines from all types of clutter objects, combine detection and classification, as well as robustness to noise, blur, and occlusion.

3.3 Cyber security

Intrusion detection is an important part of cyber security to detect malicious network activity before it compromises information availability, integrity, or confidentiality. Intrusion detection is performed using an

intrusion detection system (IDS) that classifies the network traffic as normal or intrusive. However, since normal network traffic often have similar signature as actual attacks, cyber security analysts analyze the situation for all intrusion alerts to determine whether there is an actual attack. While signature-based IDSs are often good at detecting known attack patterns, they cannot detect previously unseen attacks. Further, development of signature-based detection is often slow and expensive since it requires significant expertise. This hampers the systems adaptability to rapidly evolving cyber threats.

Many studies use ML and other AI-techniques to increase the classification accuracy of known attacks, detect anomalous network traffic (since this may indicate new attack patterns that deviate from normal net- work traffic), and automate model construction [27]. However, few of these systems are used operationally. The reason for this is that intrusion detection presents specific challenges such as lack of training data, large variability in network traffic, high cost of errors, and difficulty of performing relevant evaluations [9, 55]. Although large volumes of network traffic can be collected, the information is often sensitive and can only partially be anonymized. Using simulated data is another alternative, but it is often not sufficiently realistic. The data must then be labeled for supervised learning in terms of whether the patterns are normal or an intrusion, or for anomaly detection assured to attack-free, which is often difficult to do. Finally, the models need to be transparent so that researchers can understand the detection limits and significance of features [55].

Another measure to increase cyber security is penetration testing during security audits for identification of potentially exploitable security weaknesses. Penetration testing is often automated due to the complexity and large number of hosts in many networks. Some studies have investigated how AI-techniques may be used for simulated penetration testing using logical models of the network rather than the actual network. The network is often represented with attack graphs or trees that depict how an adversary can exploit vul- nerabilities to break into a system. However, [23] describes how models differ in terms of the way they characterize: 1) uncertainty for the attacker from abstract success and detection probabilities to uncertainty of network state, and 2) attacker actions from known pre- and post-conditions to general sensing and ob- servation of outcomes. Further, with formal models of networks and hosts, it is possible to perform what-if analysis of different mitigation strategies [5]. Future research on penetration testing will likely use cogni- tively valid models of the interaction between attacker and defender, e.g. [26], as well as deep reinforcement learning to explore the large problem space of possible attacks.

4 CHALLENGES

As indicated by the cases in Section 3, there are unsolved challenges that are important to be aware of prior to developing and deploying an AI-based application for military purposes. In this section we will discuss, in our opinion, the most critical ones for military AI: 1) transparency, 2) vulnerabilities, and 3) learning even in the presence of limited training data. Other important, but less critical, challenges related to optimiza- tion, generalization, architectural design, hyper-parameter tuning, and production grade deployment are not further discussed in this work.

4.1 Transparency

Many applications require, in addition to high performance, high transparency, high safety, and user trust or understanding. Such requirements are typical in safety critical systems [29], surveillance systems [60], autonomous agents [37], medicine [14], and other similar applications. With the recent breakthrough for AI, there is also an increased research interest in transparency to support end-users in such applications (e.g. [20, 24, 42]).

4.1.1 Expectations on transparency

The required transparency of AI depends on the end-users needs. Lipton [34] describes how transparency may concern five types of user need for:

1. Trust in situations where it is difficult for users to question system recommendations. However, it may be unclear whether user trust is based on system performance or robustness, performance relative the user, or how comfortable the user is with system recommendations.

2. Insight into previously unknown causal relationships that may be tested with other methods.

3. Knowledge of system performance limits due to limited model generalizability compared to the users abilities.

4. Some additional information about system recommendations.

5. Fairness to avoid systematic biases that may result in unequal treatment for some cases. For example, evaluation of credit applications should not be based on personal attributes, such as sex or ethnicity, although such attributes may distinguish population groups on an overall statistical level.

There are in principle, two ways to make AI-systems transparent. Firstly, some types of models are perceived as more interpretable than others, such as linear models, rule-based systems, or decision trees. In- spection of such models gives an understanding of their composition and computation. Lipton [34] describes how the interpretability depends on whether users can predict system recommendations, understand model parameters, and understand the training algorithm. Secondly, the system may explain its recommendations. Such explanations may be textual or visual. For example, by indicating what aspects of an image that mostly contributes to its classification. Miller [38] provides an extensive review of explanations in social sciences research and how this knowledge may be used to design explanations for AI systems. Typically, people explain other agents behavior in terms of their perceived beliefs, desires, and intentions. For AI systems, beliefs correspond to the systems information about the situation, desires correspond to the systems goals, and intentions correspond to intermediate states. Further, explanations may encompass abnormality of actions, preferences to minimize cost or risk, deviations from expected norms, recency of events, and controllability of actions. The major findings are that:

- Explanations are contrastive in response to particular counter-factual cases. Explanations therefore focus on why the particular recommendation was given instead of some other recommendation.
- Explanations are selected and focus on one or two possible causes and not all causes for the recommendation.
- Explanations are a social conversation and interaction for transfer of knowledge.

4.1.2 Examples of interpretable models

Bayesian rule lists (BRL) is one example of interpretable models. BRL consist of series of if (condition) then (consequent) else (alternative) statements. Letham et al. [33] describes how BRL can be generated for a highly accurate and interpretable model to estimate the risk of stroke. The conditions discretize a high-dimensional multivariate feature space that influence the risk of stroke and the consequent describes the predicted risk of stroke. The BRL has similar performance as other ML-methods for predicting the risk of stroke and is just as interpretable as other existing scoring systems that are less accurate.

Lexicon-based classifiers is another example of interpretable models for text classification. Lexicon- based classifiers multiplies the frequency of terms with the probability for terms occurring in each class. The class with the highest score is chosen as the prediction. Clos et al. [11] models lexicons using a gated recurrent

network that jointly learns both terms and modifiers, such as adverbs and conjunctions. The lexicons where trained on whether posts in forum are for or against death penalty and sentiments towards commercial productions. The lexicons perform better than other ML-methods and are at the same time interpretable.

4.1.3 Examples of feature visualization

Although DNNs offer high performance in many applications, their sub-symbolic computations with per- haps millions of parameters makes it difficult to understand exactly how input features contribute to system recommendations. Since DNNs high performance is critical for many applications, there is a considerable interest in how to make them more interpretable (see [39] for a review). Many algorithms for interpreting DNNs transform the DNN-processing into the original input space in order to visualize discriminating features. Typically, two general approaches are used for feature visualization, activation maximation and DNN explanation.

Activation maximation computes which inputs features that will maximally activate possible system recommendations. For image classification, this represents the ideal images that show discriminating and recognizable features for each class. However, the images often look unnatural since the classes may use many aspects of the same object and the semantic information in images is often spread out [43]. Some examples of methods for activation maximation are gradient ascent [13], better regularization to increase generalizability [54], and synthesizing preferred images [41, 40].

DNN explanation explains system recommendations by highlighting discriminating input features. In image classification, such visualizations may highlight areas that provide evidence for or against a certain class [68] or only show regions that contain discriminating features [3]. One approach for calculating dis- criminating features is sensitivity analysis using local gradients or other measure of variation [39]. However, one problem with sensitivity analysis is that it may indicate discriminating features that are not present in the input. For example, in image classification the sensitivity analysis may indicate obscured parts of an object rather than the visible parts [51]. Layer-wise relevance propagation avoids this problem by considering both feature presence and model reaction [4].

4.1.4 Examples of application specific explanations

In contrast to classification, AI-planning is based on models of domain dynamics. Fox et al. [15] describe how explanations for planning may use domain models to explain why actions were performed or not, why some action cannot be performed, causal relationships that enable future actions, and the need for replanning.
Since fairness is important for many AI-applications, Tan et al. [59] describe how model distillation can be used to detect bias in black-box models. Model distillation simplifies larger more complex models without significant loss of accuracy. For transparency, they use generalized additive models based on shallow trees that model each parameter and the interaction between two parameters. They train a transparent model on system recommendations from the black-box model and one transparent model on the actual outcome. Hypothesis testing of differences in recommendations from the two models shows cases where the black-box model introduce a bias, which may then be diagnosed by comparing the two transparent models. The system was evaluated on recidivism risk, lending loan risk, and individual risk for being involved in a shooting incident. The results show that one black-box model underestimates recidivism risk for young criminals and Caucasians, while overestimating the risk for Native and African Americans.

4.2 Vulnerabilities

In this section, we discuss two different aspects of vulnerabilities of DNNs: 1) vulnerability for manipulation of input and 2) vulnerability for manipulation of the model. We start by looking at manipulation of the input signal.

4.2.1 Adversarial crafting of the input

Provided a DNN, it has been found that it is easy to adjust input signal so that the classification system fails completely [58, 18, 45]. When the dimension of the input signal is large, which is typically the case for e.g. pictures, it is often enough with an imperceptible small adjustment of each element (i.e. pixel) in the input to fool the system. With the same technique used to train the DNN, typically a stochastic gradient method [49], you can easily find in which direction, by looking at the sign of the gradient, each element should be changed to allow the classifier to wrongly pick a target class or simply just misclassify. With only a few lines of code, the best image recognition systems are deceived to believe that a picture of a vehicle instead shows a dog. Figure 1 below shows the image before and after manipulation and the likelihood of the classes before and after manipulation.

The above method assumes having full access to the DNN, i.e., a so-called white-box attack. It has been found that even so-called black-box attacks, where you only have insight into the system's type of input and output, are possible [44, 56]. In [44], the authors train a substitute network using data obtained from sparse sampling of the black-box system they want to attack. Given the substitute network you can then use the white-box attack method mentioned above to craft adversarial inputs. An alternative to learning a substitute network is presented in [56], where instead a genetic algorithm is used to create attack vectors leading to misclassifications by the system. The same authors even show that it is often enough to modify a single pixel in the image, although often perceptible, to achieve a successful attack.

Figure 1: From minivan to Siberian husky. The absolute difference (amplified a factor 20) between the original and manipulated (adversarially crafted) image is shown to the right. The adversarial example (center) is generated using Kurakin's basic iterative method (BIM) described in [28].

4.2.2 Exploiting hidden backdoors in pre-trained DNNs

When designing a DNN, but only having access to a small amount of training data, it is common to use pre-trained models to achieve good performance. The concept is called transfer learning and a common procedure is to take a model that is trained on a large amount of data, replace and customize the last layers in the network to the specific problem, and then fine-tune the parameters in the final stages (and sometimes even the entire system) using the available training data. There are already a large amount of pre-trained models available for download from the Internet. A relevant question is then "How do we know that those who uploaded the model have no bad intentions?". This type of vulnerability is considered in [19] where the authors insert backdoors

into a model for recognizing US traffic signs. For example, a sticker is trained on a stop sign to belong to a class other than stop signs. They then show that a system, based the US traffic sign network, for recognizing Swedish traffic signs reacts negatively (greatly impairing the classification accuracy of the Swedish traffic sign system) when using the backdoor (i.e., placing a sticker on the traffic sign).

4.2.3 Defense methods

One way to reduce the vulnerability of the DNNs to manipulation of the input signal is to explicitly include manipulated/adversarial examples in the training process of the model [18, 28]. That is, in addition to the original training data adversarial examples are generated and used in the training of the model.

Another method is to use a concept called defense distillation [46]. Briefly described, the method tries to reduce the requirement that the output signal only point out the true class and force the other classes to have zero probability. This is done in [46] in two steps. The first step is a regular training of a DNN. In the second step, the output (class probabilities) of the first neuron network is used as a new class labels and a new system (with the same architecture) is trained using the new (soft) class labels. This has been shown to reduce vulnerability, because you do not fit the DNN too tight against the training data, and preserve some reasonable class interrelations.

Other defense methods, are for instance feature squeezing techniques such as e.g., mean or median filtering [64] or nonlinear pixel representations such as one-hot or thermometer encodings [8].

Unfortunately, neither of the methods described completely solves the vulnerability problem, especially not if the attacker has full insight into the model and the defense method.

4.3 Data

Developing ML-based applications in a military context is challenging because the data collection proce- dures in military organizations, training facilities, platforms, sensor networks, weapons, etc. were initially not designed for ML-purposes. As a result, in this domain it is often difficult to find real-world, high-quality and sufficiently large datasets that can be used to learn from and gain insight into. In this section we will explore techniques that can be used to build ML-applications even in the presence of limited training data.

4.3.1 Transfer learning

Transfer learning (also mentioned in Section 4.2.2) is a technique that is commonly used when datasets are small and when computational resources are limited. The idea is to reuse the parameters of pre-trained models, typically represented by DNNs, when developing new models targeting other, but similar, tasks. There are at least two approaches that can be used for transfer learning in DL-applications:

- Relearning the output layer: Using this approach, the last layer of the pre-trained model is replaced with a new output layer that matches the expected output of the new task. During training, only the weights of the new output layer are updated, all others are fixed.
- Fine tuning the entire model: This approach is similar to the first but in this case the weights of the entire DNN may be updated. This approach typically requires more training data.

It has been shown that transfer learning may also boost the generalization capabilities of a model. How- ever, the positive effects of transfer learning tend to decrease as the distance between the source task and target task increases [66].

4.3.2 Generative adversarial networks

Generative adversarial networks (GANs), invented by Goodfellow et al. [17], is a generative model that can be used for semi-supervised learning where a small set of labeled data is combined with a larger set of unlabeled data to improve the performance of a model [50]. The basic GAN implementation consists of two DNNs representing a generator and a discriminator. The generator is trained to produce fake data and the discriminator is trained to classify data as real or fake. When the two networks are simultaneously trained, improvements to one network will also result in improvements to the other network until, finally, an equilibrium has been reached. In semi-supervised learning, the main objective of the generator is to produce unlabeled data that can be used to improve the overall performance of the final model. GANs have, in addition to semi-supervised learning, also been used for:

- Reconstruction: Filling the gaps of partly occluded images or objects [65].
- Super-resolution: Converting images from low resolution to high resolution [32].
- Image-to-image translation: Converting images from winter to summer, night to day, etc. [67]. A military application of this technique could be to convert night-vision images to daylight images.

4.3.3 Modeling and simulation

Modeling and simulation has been used extensively by the military for training, decision support, studies, etc. As a result, there are lots of already validated models that have been developed over long periods of time that could also potentially be used to generate synthetic data for ML-applications. As an example, a flight-simulator could be used to generate synthetic images of aircrafts placed in different environmental settings. Labeling is in this case automatic since the aircraft type is known prior to generating the synthetic image. However, not surprisingly, using synthetic images may result in poor performance when applying the model to real-world images. One approach that is currently being explored is to enhance the synthetic image using GANs to make it photo-realistic. This approach was successfully applied in [53].

5 CONCLUSIONS

The recent breakthrough of AI is gradually reaching a point where it can be used in military applications. The paper describes some possibilities for using AI in surveillance, underwater mine warfare, and cyber se- curity. Other potential applications are reconnaissance using partly autonomous vehicles and sensor systems, threat evaluation in air defense systems with high temporal requirements, intelligence analysis of emerging patterns, command and control systems, and education and training. However, military applications of AI need to consider challenges in terms of:

- Transparency to assure model performance that is consistent with military requirements.
- Vulnerabilities that may drastically reduce system performance.
- Insufficient training data for ML.

Many advancements have already been made by researchers focusing on the transparency, interpretabil- ity, and explainability issues of AI. Many of these advancements can likely also be used in military AI-applications. However, a more thorough requirements analysis is needed to understand how to utilize these research results. Military requirements may be very different regarding risk, data quality, legal demands, etc. and some types of transparency may not even be applicable. Further, more studies are needed on how to utilize social science research to improve AI-explainability. Future studies should also include how to utilize the rich set of visualization techniques that are developed in the visual analytics research area.

Since there is currently no silver bullet for the vulnerability problem, it is important to monitor this research area and continuously look for promising solutions. However, until such solutions are available it is necessary to minimize external access to models and defence techniques. Opponents may otherwise try to utilize the vulnerabilities to their advantage.

Finally, transfer learning makes it possible to adapt pre-trained models to military applications where there is both limited training data and computational resources. GAN is another promising technique that enables learning using labeled and unlabeled data (semi-supervised learning). GAN can also be used in combination with simulation to improve the realism of synthetically generated training data.

ACKNOWLEDGMENT

This work was supported by the FOI research project “AI for decision support and cognitive systems”, which is funded by the R&D programme of the Swedish Armed Forces.

REFERENCES

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, and Carl Case et al. Deep speech 2 : End-to-end speech recognition in english and mandarin. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 173–182, New York, New York, USA, 20–22 Jun 2016. PMLR.

[2] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. Lipnet: End-to-end sentence-level lipreading. *GPU Technology Conference*, 2017.

[3] Housam Khalifa Bashier Babiker and Randy Goebel. Using KL-divergence to focus deep visual explanation. *arXiv preprint arXiv:1711.06431*, 2017.

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[5] Michael Backes, Jörg Hoffmann, Robert Künnemann, Patrick Speicher, and Marcel Steinmetz. Simulated penetration testing and mitigation analysis. *arXiv preprint arXiv:1705.05088*, 2017.

[6] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. pages 153–160. International Conference on Neural Information Processing Systems, NIPS, 2006.

[7] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

[8] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: one hot way to resist adversarial examples. International Conference on Learning Representations, ICLR, 2018. https://openreview.net/pdf?id=S18Su–CW.

[9] Carlos A Catania and Carlos GarcíA Garino. Automatic network intrusion detection: Current techniques and open issues. *Computers & Electrical Engineering*, 38(5):1062–1072, 2012.

[10] Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[11] J Clos, N Wiratunga, and S Massie. Towards explainable text classification by jointly learning lexicon and modifier terms. In *IJCAI-17 Workshop on Explainable AI (XAI)*, pages 19–23, 2017.

[12] Killian Denos, Mathieu Ravaut, Antoine Fagette, and Hock-Siong Lim. Deep learning applied to underwater mine warfare. In *OCEANS 2017-Aberdeen*, pages 1–7. IEEE, 2017.

[13] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1–13, 2009.

[14] John Fox, David Glasspool, Dan Grecu, Sanjay Modgil, Matthew South, and Vivek Patkar. Argumentation-based inference and decision making–a medical perspective. *IEEE intelligent systems*, 22(6), 2007.

[15] Maria Fox, Derek Long, and Daniele Magazzeni. Explainable planning. *arXiv preprint arXiv:1709.10256*, 2017.

[16] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. pages 249–256. International Conference on Artificial Intelligence and Statistics, AISTATS, 2010.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Process- ing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[18] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversar- ial examples. pages 1–11. International Conference on Learning Representations, ICLR, 2015. https://arxiv.org/abs/1412.6572.

[19] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. https://arxiv.org/abs/1708.06733, 8 2017.

[20] David Gunning. Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA)*, 2017.

[21] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computing*, (18):1527–1554, 2006.

[22] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. https://arxiv.org/abs/1207.0580, 7 2012.

[23] Jörg Hoffmann. Simulated penetration testing: From “dijkstra” to “turing test++”. In *ICAPS*, pages 364–372, 2015.

[24] IJCAI. Workshop on explainable artificial intelligence (XAI). 2017.

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Learning Representations, ICLR, 2015. https://arxiv.org/abs/1502.03167.

[26] Randolph M Jones, Ryan OGrady, Denise Nicholson, Robert Hoffman, Larry Bunch, Jeffrey Bradshaw, and Ami Bolton. Modeling and integrating cognitive agents within the emerging cyber domain. In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, volume 20. Citeseer, 2015.

[27] Gulshan Kumar, Krishan Kumar, and Monika Sachdeva. The use of artificial intelligence based techniques for intrusion detection: a review. *Artificial Intelligence Review*, 34(4):369–387, 2010.

[28] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. https://arxiv.org/abs/1611.01236, 11 2016.

[29] Zeshan Kurd, Tim Kelly, and Jim Austin. Developing artificial neural networks for safety critical systems. *Neural Computing and Applications*, 16(1):11–19, 2007.

[30] Rikard Laxhammar. Anomaly detection for sea surveillance. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008.

[31] Rikard Laxhammar, Goran Falkman, and Egils Sviestins. Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 756–763. IEEE, 2009.

[32] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 105–114, 2017.

[33] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[34] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[35] L. J. Luotsinen, F. Kamrani, P. Hammar, M. Jändel, and R. A. Løvlid. Evolved creative intelligence or computer generated forces. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 003063–003070, Oct 2016.

[36] Steven Mascaro, Ann E Nicholso, and Kevin B Korb. Anomaly detection in vessel tracks using bayesian networks. *International Journal of Approximate Reasoning*, 55(1):84–98, 2014.

[37] Joseph E Mercado, Michael A Rupp, Jessie YC Chen, Michael J Barnes, Daniel Barber, and Katelyn Procci. Intelligent agent transparency in human–agent teaming for multi-uxv management. *Human factors*, 58(3):401–415, 2016.

[38] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *arXiv preprint arXiv:1706.07269*, 2017.

[39] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

[40] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.

[41] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

[42] NIPS. Workshop on explainable artificial intelligence (XAI). 2017.

[43] Fabian Offert. "I know it when i see it". Visualization and intuitive interpretability. *arXiv preprint arXiv:1711.08042*, 2017.

[44] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. pages 1–14. ASIA CCS'17, 2017. https://arxiv.org/abs/1602.02697v4.

[45] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. pages 1–11. IEEE European Symposium on Security & Privacy, 2016. https://arxiv.org/abs/1511.07528v3.

[46] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy*, pages 582–597. IEEE, 2016.

[47] Bradley J Rhodes, Neil A Bomberger, Michael Seibert, and Allen M Waxman. Maritime situation monitoring and awareness using learning mechanisms. In *Military Communications Conference, MILCOM*, pages 646–652. IEEE, 2005.

[48] Bradley J Rhodes, Neil A Bomberger, and Majid Zandipour. Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness. In *Information Fusion, 2007 10th International Conference on*, pages 1–8. IEEE, 2007.

[49] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, (9):533–536, 1986.

[50] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.

[51] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[52] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. *CoRR*, abs/1712.05884, 2017.

[53] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. *CoRR*, abs/1612.07828, 2016.

[54] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[55] Robin Sommer and Vern Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 305–316. IEEE, 2010.

[56] Jiawei Su, Danilo V. Vargas, and Sakurai Kouichi. One pixel attack for fooling deep neural networks. https://arxiv.org/abs/1710.08864, 10 2017.

[57] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph.*, 36(4):95:1–95:13, July 2017.

[58] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. International Conference on Learning Representations, ICLR, 2014. https://arxiv.org/abs/1312.6199.

[59] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Detecting bias in black-box models using transparent model distillation. *arXiv preprint arXiv:1710.06169*, 2017.

[60] Michael Tom Yeh et al. Designing a moral compass for the future of computer vision using speculative analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 64–73, 2017.

[61] Kenneth Tran, Xiaodong He, Lei Zhang, Jian Sun, Cornelia Carapcea, Chris Thrasher, Chris Buehler, and Chris Sienkiewicz. Rich image captioning in the wild. *CoRR*, abs/1603.09016, 2016.

[62] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.

[63] David P Williams. Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2497–2502. IEEE, 2016.

[64] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing mitigates and detects carlini/wagner adversarial examples. http://arxiv.org/abs/1705.10686, 5 2017.

[65] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3d object reconstruction from a single depth view with adversarial learning. In *International Conference on Computer Vision Workshops (ICCVW)*, 2017.

[66] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

[67] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

[68] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.