Deep Learning Application in Security and Privacy – Theory and Practice: A Position Paper

Julia A. Meister, Raja Naeem Akram, and Konstantinos Markantonakis

Information Security Group, Smart Card and IoT Centre,
Royal Holloway, University of London.
julia.a.meister@gmail.com, {r.n.akram, k.markantonakis}@rhul.ac.uk

Abstract. Technology is shaping our lives in a multitude of ways. This is fuelled by a technology infrastructure, both legacy and state of the art, composed of a heterogeneous group of hardware, software, services, and organisations. Such infrastructure faces a diverse range of challenges to its operations that include security, privacy, resilience, and quality of services. Among these, cybersecurity and privacy are taking the centre-stage, especially since the General Data Protection Regulation (GDPR) came into effect. Traditional security and privacy techniques are overstretched and adversarial actors have evolved to design exploitation techniques that circumvent protection. With the ever-increasing complexity of technology infrastructure, security and privacy-preservation specialists have started to look for adaptable and flexible protection methods that can evolve (potentially autonomously) as the adversarial actor changes its techniques. For this, Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) were put forward as saviours. In this paper, we look at the promises of AI, ML, and DL stated in academic and industrial literature and evaluate how realistic they are. We also put forward potential challenges a DL based security and privacy protection system has to overcome. Finally, we conclude the paper with a discussion on what steps the DL and the security and privacy-preservation community have to take to ensure that DL is not just going to be hype, but an opportunity to build a secure, reliable, and trusted technology infrastructure on which we can rely on for so much in our lives.

Keywords: Security · Privacy · Machine Learning · Deep Learning · Application.

1 Introduction

Computing technology is an integral part of our lives and has many facets ranging from supercomputing (used in weather prediction, cutting-edge research, and business automation) to embedded devices (like smartphones, electronic devices in a home, and intelligent transport systems). Among many, security and privacy are considered to be two distinct and unique challenges. In the security and privacy domain, any protection system has to match a constantly evolving adversarial actor. According to the Symantec cybercrime report [1], the overall

number of vulnerabilities has increased by 13% in 2018. Similarly, according to Cybersecurity Ventures [2], zero-day exploits seen in the wild will grow from one per week (in 2015) to one per day by 2021. It is practically impossible for a human to keep pace with the sheer number of cybersecurity events (and related activities) on a daily basis on top of an already daunting threat landscape [3].

In this paper, and as a matter of fact in any context, security and privacy are relative terms. It is not discussed as an absolute state, but rather as a state with potential and/ or accepted risks. The global cost of data breaches has increased by 6.4% [4] and has the potential to severely damage an organisation's bottom-line, even without taking the potential penalties imposed by the General Data Protection Regulation (GDPR) into account [5]. As per the GDPR, an organisation can be fined up to €10 million or 2% of the firm's global turnover for a small offence (whichever is greater). For a serious offence, an organisation can be fined up to €20 million or 4% of a firm's global turnover (whichever is greater) [5].

Furthermore, there is a crisis of skilled cybersecurity practitioners. According to Ciccone, the cybersecurity job market will grow by approximately 6 million USD globally by 2019 – with potential shortages of trained professionals up to 25% [6]. Automation of decisions and actions based on network and system generated alerts has the potential to help overcome the challenges related to security and privacy – both in a technological and a business-operations (e.g. labour shortages) dimension.

Artificial Intelligence (AI) is seen as a potential solution towards the cybersecurity automation challenge in some academic and industrial circles. Machine Learning (ML) has been successfully deployed in a number of domains including but not limited to: image classification [7], object detection and recognition [8], language translation, and voice synthesis [9]. In many cases, Deep Learning (DL), a type of Machine Learning (ML) method, does not require prior expert knowledge for its learning (an obvious exception is Neuro-Fuzzy techniques). Therefore, it generally needs less manually engineered feature extraction and specialist knowledge [10]. DL can detect patterns in raw data by transforming it into higher and more abstract level representations - a function that is very interesting for cybersecurity zero-day vulnerability/ exploit detection. Similarly, DL is used to abstract malware's behavioural features and anomalous activities and can then be used to detect its existence in a system [11, 12].

AI as a cybersecurity tool is expected to capture a large market and it is clear that AI has the potential to impact the cybersecurity space [13]. Furthermore, there is sufficient market interest in both commercial (financial incentives) and academic research. In this paper, we discuss the challenges of deploying AI-based techniques (ML/ DL) to security domains as a general security tool and highlight the difference between the theory and practice. The discussed challenges come from the technical development and exploration of DL methods in the context of cybersecurity – showcasing the fact that DL techniques in themselves are not the panacea but mearly a tool that requires a number of correct (and in some cases trustworthy) features to be effective. It is understood that there is a potential

to mislead an ML/ DL deployment as discussed in existing literature [14, 15], which is not the focus of this paper.

The robustness of DL is stated in [14] as inversely proportional to the potential of an attacker's ability to find adversarial examples, which can impact the accuracy of detection and classification of a threat. However, we argue that robustness, no doubt an important feature, is not just dependent on the attacker's ability to find adversarial examples. It is also affected by the interdependent relationship of input data, its accuracy and trustworthiness, its feature-richness, how representative the data is of all possible case scenarios, and the potential of adversarial samples. We will discuss these features in further detail throughout the paper. Furthermore, we examine ML/ DL not only from the view of theoretical and feature/ ability specific limitations but also from the view of practical challenges related to implementation and deployment. For the most part, existing papers focus on discussing a specific model's success rate and implementation/ deployment challenges. They do not include discussions on the general challenges related to ML/ DL deployed as security and privacy mechanisms.

1.1 Structure of the Paper

Section 2 elaborates on the existing academic work that has shown the promise of ML/ DL as an automation tool for security and privacy practices. In Section 3, we dive into the technical discussion of DL and how automation based on it is designed and developed. The discussion is derived from the author's first impressions and practical experience coming from a security background. Section 4 articulates the practical considerations that a security practitioner has to take into account when working on DL deployment. Section 5 is a list of DL features that would make the technology a useful security tool for cybersecurity practitioners.

2 Security and Privacy by Deep Learning

In this section, we survey the types of security and privacy services and applications in which DL has been deployed successfully – as represented by academic literature.

2.1 Deep Learning for Security and Privacy

The set of security and privacy services that are being explored in academic literature to be the target deployment scenarios for DL are as follows:

1. *Malware Detection*: Efficient pattern recognition in large datasets is what ML/ DL is purpose built for. A number of proposals are put forward in academic literature that identify malware with high accuracy [16, 17]. In most of these proposals, pattern recognition is based on a particular behaviour (communication, syscall and resource usage/ utilisation patterns, etc). For an adversarial entity, the objective is to hide or exhibit its behaviour within the scope of genuine applications to avoid detection.

2. *Anomaly Detection or Network Intrusion Detection*: Both anomaly and network intrusion detection by ML/ DL rely on network traffic analysis to find usage and communication patterns that represent an abnormal behaviour. It is important to keep in mind that anomalous behaviour is not necessarily a set of activities that are prohibited by system policies (security/ privacy). It is just an out-of-the-ordinary activity that can be genuine or malicious. For example, user A has access to client records. Usually, user A only accesses one record a day, but today user A accesses the entire list of clients. If the access control policy only focused on access (may user A access client records?) and not on frequency (how many client records can user A access?), accessing all client records would be a permitted action and not suspicious. However, this action might be anomalous. Such classification and detection of out-of-pattern usages fits nicely within the current capabilities of ML/ DL technology [18]–[20].
3. *Distributed Denial of Service (DDoS) Detection*: DDoS can be viewed as an anomalous request to access a particular resource. Therefore, ML/ DL can efficiently identify out-of-pattern access requests based on the access patterns to a particular resource (e.g. a website or an application) [21, 22].

From the above list, we can ascertain that DL is not widely used for privacy-preservation techniques. There is a potential for exposing data on user access patterns based on the user connection graph, especially in the context of data flow analysis. These domains might have unique patterns that can be useful for an effective DL deployment but an academic literature search for applications of DL in these fields did not yield substantial results. Below, we explain some of the identified privacy related services that might be suitable for DL deployment but limited work has been carried out in academic literature:

1. *Data Flow Analysis*: The flow of data between any two entities can reveal data consumption in an organisation. For example, the flow of data between the consumer database and marketing teams can represent potential value to consumer profiling, targeted marketing, and campaign analysis. The data flow and usage in a specific enterprise have a set pattern, even when only looking at individual features such as 'data flow' and the actual 'contents of the data'. Therefore, ML/ DL can be used to identify anomalies in the usage of data based on its analysis, and the resulting anomalous data flow patterns could be very useful for an Intrusion Detection System (IDS) or Intrusion Detection Prevention (IDP) but not as a privacy preservation function.
2. *Data Exposure Potential*: Whether in an enterprise environment or in personal settings, individuals have a circle of other individuals with whom they communicate. A community map for each individual can be constructed based on these communication patterns which can represent not only 'with whom' individuals share information but also 'what information' is being shared with their community. For example, an individual shares one type of information with only a subset of the individuals in his/ her community. This is easily classifiable and based on the patterns, ML/ DL can predict whether

information accessible to an individual at a particular point in time has a high probability of being shared with certain other individuals. This analysis can be used to build a data exposure prediction which could be a useful tool for privacy-preservation and assessment. Furthermore, in the event of an information leakage, an analysis of the data flows and the probability of data exposure can be incorporated into the forensic investigation to quickly find any potential points (individuals) that could have leaked the information. The potential of ML/ DL has not been fully explored in the context of data exposure in current academic literature. We believe that the application of ML/ DL for such analysis shows a lot of promise.

Most of the existing literature about privacy and DL is focused on how to design DL methods in a manner that does not violate the users' privacy [23]–[25]. Another application of DL in privacy is to build recommendation systems for users. For example, Yu et al. [26] put forward a privacy setting recommendation system (iPhoto) for photo sharing based on image analysis. Most dimensions related to DL and privacy are beyond the scope of the this paper. The scope of the paper is how DL itself can be used as a privacy-protection mechanism.

3 Deep Learning - A Deeper Look at its Application

In this section, we explore the technical aspects of understanding and deploying DL. The discussion revolves around the pre-requisites for DL deployment, the tools that can be used, and DL optimisation. Readers are referred to consult the survey by Zubair et al. [27] for an in-depth analysis of DL structures and methodologies.

3.1 Representation Learning

DL uses representation learning algorithms to automatically identify complex hidden structures in large datasets [10]. Relations between parameters can be more or less hidden depending on the features present in the data. Representation learning works to solve this problem by transforming raw data into a more useful representation for detection and classification predictors by highlighting the important dependencies [28]. The challenge is to generalise as much as possible while also preserving most of the information in the original dataset.

DL implements the learning technique in the form of a model, a concatenation of multiple, relatively simple layers that each perform a transformation on the data [28]. The layers' input is either raw data (input layer) or the previous layer's learned representation of its input (hidden and output layers). This leads to automatically identified, hierarchical levels of abstraction, also called feature extraction, with higher level features defined as a composition of lower-level features [29, 30]. During the training phase, the model adjusts the internal parameters used to transform the data to achieve a more useful result [10].

3.2 Data Normalisation

DL models rely heavily on data as it is the basis of the pre-training and training phases, which in turn underlie the specialisation of a model to a task.

DL does not need a perfectly curated dataset due to its learning scheme. Semi-supervised techniques have been shown to alleviate problems, however, a new training strategy and a better cost function could make training on incomplete and noisy data sets more efficient [31]. Whitening data is a known way of speeding up training convergence, readers are referred to [32] for details on how to transform the input data.

Ioffe and Szegedy [33] describe batch normalisation, where normalisation is embedded in the model architecture as another method to reduce training-times. It works towards fixing the distribution of the layer's inputs and thereby solves the problems introduced by internal covariate shift. Internal covariate shift describes the fact that the layers' input distribution continuously changes during training due to the internal parameters updating [33]. The difficulty in changing the dataset in any way is to preserve as much of the original information as possible. This can be achieved by normalising the training examples relative to the entire training data [33]. Other, less efficient ways of combating covariate shift include lowering the training rate and careful parameter initialisation. Using DL in combination with Big Data is a popular concept in the industry, however, there are many challenges that need to be overcome. The three V's model identifies them as volume, variety, and velocity.

Chen and Lin [31] provides the authors' thoughts on how to solve these problems. According to the authors, the large volume of Big Data (number of inputs, number of represented classes and high dimensionality of the entries) cannot be accommodated by a single machine due to its limited memory and computing capacity. A distributed framework would be more suited to the task. DL has been successfully utilised for the integration of heterogeneous data, e.g. [34] and [35]. Therefore, the authors believe that DL methods can be applied to Big Data's large variety of data structures with further optimisation work. They propose online learning to combat the velocity (how quickly data is generated).

There are many large datasets ranging across a wide selection of categories publicly available which can be used in training and testing networks. Examples are the MNIST database[1] of handwritten digits and the Google Audioset[2], which includes thousands of labelled audio clips. Kaggle[3] is a platform that hosts ML competitions and maintains public datasets.

3.3 Designing Deep Learning Models

There are different *neural network architectures* used in DL, each with their own advantages and disadvantages. Convolutional networks are a type of feedforward

[1] `http://yann.lecun.com/exdb/mnist`
[2] `https://research.google.com/audioset`
[3] `https://www.kaggle.com`

network that are designed to process multidimensional signals such as images and video [36], whereas recurrent networks are adapted to work with sequence data which makes them more difficult to train but applicable to natural language processing (NLP) challenges [37]. Deep Belief Networks (DBNs) are made up of several layers of restricted Boltzmann Machines (RBMs) and are useful for when the training data set is made up of both labelled and unlabelled entries. They often perform better than networks trained only with backpropagation [36].

The *training distribution and structure* can be an important factor in the choice of model and learning method. Supervised learning methods require labelled data and tend to have good results when large quantities of data are available [29]. They adjust the model's internal parameters based on the training loss, calculated by comparing the predicted output to the expected output as defined by the data entry's label. When it comes to unsupervised learning, the ultimate goal is to abstract the raw data in a way that identifies the important factors of variation that apply to all classes. Bengio has had success applying a transductive strategy by using linear models such as Principal Component Analysis (PCA), among others, as some of the network's layers [30]. Semi-supervised learning makes use of both labelled and unlabelled data. The RBMs that make up a DBN are pre-trained with an unsupervised greedy layer-by-layer algorithm and the whole model is then fine-tuned with labelled data and backpropagation. DBNs often perform better than networks trained solely with backpropagation [36], as the combination of non-linear layers in a model can be sensitive to the initialisation values. Pre-training, as used with DBNs, can mitigate this sensitivity [29].

When it comes to optimising a model's accuracy, tuning the *hyperparameters* is an important step. They are values that directly influence the training of a neural network by configuring a model's complexity and the learning process [38], both of which are critical to the model's performance. However, finding the ideal values for these parameters can be very difficult as fine-tuning is often based on experience. According to Bengio, there are two common ways of optimising a model's performance through the choice of hyperparameters: manual trial and error and a grid search. Both approaches run into problems when the number of parameters is too large [30]. Readers are referred to [30] and [39] for a more efficient optimisation based on random search and greedy exploration. The number and type of parameters differ between models and learning algorithms. Some of the most common include the learning rate, momentum, number of hidden units, number of epochs and batch size.

Training large, distributed networks is slow, as the use of parallel resources is very inefficient. Denil et al. introduce a way to reduce the number of free parameters without dropping the accuracy, as many parameters can be predicted and are, therefore, redundant [40].

Over- and underfitting describe situations where a neural network has not learned the ideal generalisation of the training data which leads to poor performance when new data is introduced. This can also be described as the bias/variance dilemma, a trade-off between high bias and high variance [41]. Common

metrics such as training and test error are used to analyse the accuracy of a model can help identify over- and underfitting.

High variance means that a model fails to differentiate between the signal (the general, underlying pattern) and the noise (dataset-specific randomness) of a dataset. In other words, an *overfit model* has failed to sufficiently generalise the features of its specific training distribution and therefore performs poorly on previously unseen data, as it has no general knowledge it can apply. Overfitting can occur with a complex model whose learning algorithm has a low bias and a high variance. Cross validation is a proven method of preventing overfitting by stopping training before the specification becomes to high [42]. The point in time at which to stop training is identified by comparing the model's accuracy on the training data to its accuracy on the unseen testing data. Training is stopped if the difference starts growing or is deemed too large, also called early stopping. Reducing the number of parameters is another method of combating overfitting [42]. Dropout layers have also been shown to be successful because they prevent the co-adaption of a network's hidden units [29]. They introduce unpredictable noise into the data by dropping random parameters in each training iteration.

Bias describes the difference between the model's expected output and the correct values. High bias occurs when the model is oversimplified and does not have enough flexibility to capture the underlying relations of features present in the data or when there are insufficient parameters. A model is said to be *underfit* if it has a low variance but a high bias and this can be identified by a high error on both the training and the test data. A possible solution to this problem is changing the model's structure and parameters so that it better fits the problem to be solved.

Bias and variance are inversely related. The ideal model minimises the expected total error of a learning algorithm, which is defined as the sum of squared bias, variance and irreducible error. While bias and variance are reducible, the irreducible error comes from modelling the problem itself.

3.4 Deploying Deep Learning Methods

There are many open-source tools and frameworks that support DL which can vary greatly in overhead, running speed and number of pre-made DL components. Following are short descriptions of a small selection of them.

TensorFlow[4] is a Python-based library with automatic differentiation capabilities that supports both ML and DL. The high-performance numerical computations, modelled as data flow graphs, can be applied to other domains as well. TensorFlow is used by companies such as Google, Uber, and AMD.

PyTorch[5] is another such library which enables rapid research on ML networks. The focus lies on extensibility and low overhead, which is possible because the core logic is written in C++. It also supports reverse mode automatic differentiation,

[4] `https://www.tensorflow.org`
[5] `https://pytorch.org`

which is the most important type of differentiation for DL applications [43] and distributed training. In 2017, Uber AI Labs released Pyro[6], a deep probabilistic programming language (PPL) based on PyTorch.

Caffe[7] is a C++ library that provides interfaces for Python and MATLAB [44]. It is a clean and modifiable framework, due to the fact that the model's representation is separate from the model's implementation [45]. It is very fast in training convolutional networks and allows for seamless switching between devices (CPU and GPU).

MATLAB[8] can be used for DL among other things and allows users to build and analyse models, even with little expert knowledge in DL. It provides access to models such as GoogLeNet and AlexNet and is compatible with models from Caffe and TensorFlow-Keras. MATLAB also supports collaboration with the PyTorch and MXNet frameworks.

MXNet[9] is a very versatile DL framework which supports imperative and symbolic programming as well as multiple languages, such as C++, Python, R, Scala, MATLAB and JavaScript. Its running speed is similar to Caffe and significantly faster than TensorFlow. It is used by both AWS and Azure, among others [44].

4 Practical Considerations of Deep Learning Deployment

In this section, we discuss the challenges related to deploying DL as part of the cyber security and privacy-preservation mechanism. We discuss three major issues related to the DL, which is in no way an exhaustive list. However, the problems listed in this section have a significant impact on current DL implementations.

4.1 Training Data Set

Any DL technique requires training to achieve specialisation for a task, therefore the training data set and its structure are very important. There are two crucial elements about the training data set: a) feature-richness and b) trustworthiness.

Feature-richness means that the training data should include an extensive collection of information so that the DL model can identify as many features as possible, which will help it differentiate between genuine and malicious behaviours accurately once it is deployed. Features have to be as extensive as possible; For example, data related to an activity should cover as much information about that activity as possible so a malicious entity has as very little room to manoeuvre and trick the deployed DL system. Furthermore, the training data should include a diverse set of behaviours. If a training data set is representative of a behaviour set, the algorithm has a better chance of accurately classifying features in it. If the behaviour set is not comprehensive, any behaviour that is not part of the

[6] `http://pyro.ai`
[7] `http://caffe.berkeleyvision.org`
[8] https://uk.mathworks.com
[9] `https://mxnet.apache.org`

set might be miscategorised because the DL model could fail to differentiate between a genuine and malicious behaviour correctly. This failure is due to the fact that DL builds its knowledge base of genuine and malicious behaviour from the training dataset during the training phase. One of many learning techniques is re-enforced learning. Many learning techniques can open up a potential avenue for an adversary to modify the behaviour classification of an ML/ DL system.

The second crucial element is the data's trustworthiness. As one of the most important elements of DL, data should be sourced from a trusted environment and this is also true for malicious activities captured (and tagged) for the training data set. The challenge is to capture malicious activities in a trusted manner from a real environment or a lab simulation that accurately depict how an attacker could behave. As a note, training is carried out on a data set that represents 'past' attacks (known attack patterns) and will not necessarily be representative of 'future' attacks (unknown vulnerability and attack patterns). The challenges related to new and unknown attacks are further discussed in Section 5.

4.2 Adversarial Samples

There is extensive work in academic literature that discusses the impact and limitation of ML/ DL against adversarial samples [46]. From a deployment point of view, security and privacy practitioners have to keep in mind that a deployed DL system can be susceptible to adversarial samples. This means that an attacker could influence the DL model's training to learn malicious activities as genuine. By doing so, attackers are enabled to accomplish their goal without DL detecting and flagging them. The challenge related to adversarial samples is crucial, as organisations deploying DL based security and privacy mechanism would prefer for them to evolve over time, thereby accommodating the increasing sophistication in the threat landscape. However, allowing the evolution of a DL model after initial training opens it up to adversarial samples. On the other hand, a DL technique restricted to the initial training is not flexible and extensible, two of the important functions DL needs to cope with the challenges of cybersecurity and privacy. A potential middle ground could be to select a DL technique that is the least susceptible and designed to withstand adversarial samples. Unfortunately, even with such methodologies, the likelihood of adversarial samples cannot be completely removed. Therefore, adversarial samples are a threat vector that will see more sophistication in the future as more and more organisations deploy ML/ DL based cybersecurity and privacy-preservation mechanisms.

4.3 General Data Protection Regulation (GDPR)

Organisations dealing with EU citizens' data have to comply with GDPR regulations. GDPR gives a number of rights to consumers, among which are the two that we are going to discuss in this section: Right-to-Know (RtK) and Right-to-Rectification (RtR).

When it comes to processing user data, the Right-to-Know (RtK, Article 15.1.h) states that data subjects have the right to know about the "the existence

of automated decision-making, including profiling, referred to in Article 22(1) and (4) and, at least in those cases, meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject" [5]. This article requires the availability of meaningful information about the processing method used to process users' data. As discussed before, DL is chaotic in many instances and the steps taken to reach a particular decision might have limited traceability or support for reverse-engineering. As an example, a user is in his or her rights to request information on why they received a certain result from an organisation. The organisation then has to explain how the user's data was processed by the company's AI to generate that particular result. GDPR also holds firms accountable for bias and discrimination in their automated decisions. The challenge of explaining how DL has reached a specific decision becomes paramount – an aspect of DL that has not been extensively investigated. To what extent DL's choice can be explained and whether that is an acceptable and, more importantly, meaningful explanation to the regulatory-authorities and consumer needs to be further researched.

The Right-to-Rectification (RtR, Article 16) states that "[t]he data subject shall have the right to obtain from the controller without undue delay the rectification of inaccurate personal data concerning him or her" [5]. If a user exercises RtR, they request changes to their personal data stored in the system. How these change in the data will impact previous processing and leaning, which are now based on incorrect data, is still a big question. The challenge is to make DL rectify its input data selectively post-processing in a manner that does not require a complete retraining.

On a side note, depending on how DL is deployed, the Right-to-Forget (RtF, GDPR Article 17) might have an impact if a sufficient number of consumers/ users request their data to be deleted. At that point, the knowledge set reflecting the behaviour of an organisation's consumers/ users will not be accurate anymore. How this impacts DL's subsequent decisions is still unclear and requires further investigation.

As a cyberseucrity and privacy practitioner, a clear view of the needs and visions for a DL deployment are necessary. There are plenty of unanswered questions related to DL in terms of research (Section 5), operation, and legislation (GDPR). It is safe to say that this technology has the potential to be beneficial by improving security and privacy-preservation. However, the pertinent question is whether it is ready and mature enough to be deployed extensively as a security and privacy mechanism. The answer to this is complex and depends on multiple factors, including:

1. Organisational requirements and the prioritised security objectives.
2. How the organisation envisions using ML/ DL, keeping in mind that ML/ DL are not silver bullets.
3. Understanding the limitations of ML/ DL and complimenting these techniques with traditional security and privacy measures.
4. Accepting that ML/ DL are in the early stage of development and might go through many developments and improvements in the next few years,

therefore deployed systems will have to keep up with rapid change (flexibility, extensibility, and scalability).

5 Research Challenges for Deep Learning

In this section, we put forward list of relevant topics and questions for ML/ DL research from the perspective of a cybersecurity practitioner.

1. *Policy change impact analysis*: In an enterprise environment, policies change regularly, and can be related to the security and privacy aspects of the enterprise. The impact assessment of such policies on the enterprise environment is based on human experts' knowledge. If the enterprise has deployed ML/ DL as a security and privacy measure, policy changes need to be reflected in the ML/ DL method's learning and execution. To the authors' knowledge, there is no evaluation of how dynamic policies will impact currently deployed ML/ DL implementations. Therefore, predictive impact analysis of policy changes on DL based security and privacy mechanism would be a important step forward.
2. *Defining a new policy*: An organisation's security and privacy objectives are specified by policies and rule-sets. In existing DL, these policies and rule-sets are represented in the labelling of individual records in the training data set. If the policy changes after the deployment of a DL based system, the available option is to generate a new training dataset based on the new policies and retraining the DL model. Generating the training data set and retraining can be considered costs in terms of performance and time. The challenge is to cut down this cost and make policy changes as similar to traditional security mechanisms like firewall, access control and IDS, to name a few.
3. *Preparing DL to cope with the 'future'*: The cybersecurity and privacy landscape is constantly evolving. To cope with this change, DL has to be flexible and have the ability to learn new patterns even after deployment. Furthermore, prior knowledge already learned by a particular instance of DL is valuable, and the ability to transfer it to other instances (for example among multiple organisations) would vastly improve the readiness of the collective cybersecurty field. A potential path forward could be to develop DL techniques with lifelong learning capabilities.
4. *Isolated or Collaborative Learning*: Isolated learning has its pros and cons. The positive side is that as an organisation, the training set will include behaviours specific to your organisation. However, this also means that unless you experience a cyber attack, you will not be able to profile it. With collaborative learning, if a single instance of the collaboration experiences a cyber attack, its profiling can then be shared with the other instances in the group. This has the potential to rapidly improve security countermeasures against new and previously unknown attacks. Collaborative learning introduces some additional challenges, such as:
   - *Knowledge based collaboration*: In collaborative learning, should algorithms share their learned knowledge or simply share the raw records

of the out-of-profile observations? It also requires a method for sharing prior knowledge between multiple DL instances.
- *Raw records based collaboration*: Sharing raw records seems simple, as each instance can run its own learning process over it. However, this could leak security sensitive data and violate privacy requirements. For raw records based collaboration, efficient and strong anonymisation techniques have to be developed. This anonymisation technique has to protect privacy and security sensitive data but at the same retain sufficient features so that it is still useful for training other DL instances.

5. *Making deep learning forget*: There are a number of situations where it is preferable to make the DL de-profile specific records from its knowledge base. For example, a) the discovery of malicious data in the training data set that is now required to be re-labelled as malicious, b) removing adversarial samples from the DL knowledge and c) if a consumer/ user exercises RtR (Right-to-Rectification) or RtF (Right-to-Forget) under GDPR. In such situations, DL techniques need to 'forget' about certain records. How to achieve this seems to be an open question that will be crucial in a future with increased awareness about privacy in the general public and adversaries successfully training DL implementations with adversarial samples.

6 Conclusion

In this paper, we briefly explore the potential, practicality, implications, and shortcomings of DL mechanisms in fields such as security and privacy preservation mechanisms. There are numerous proposals in academic literature that advocate the success of DL as an effective mechanism for cybersecurity. We do not evaluate their claims in this paper. We view DL as a mature domain and evaluate how a security practitioner would go about deploying it, what challenges and issues they would have to overcome, and what options are available to resolve some of these issues. We are of the opinion that DL has come a long way and can potentially be applied to security and privacy functions with a defined set of static behaviours. In such situations, DL can efficiently detect any behavioural violations with high accuracy. However, it is too early to consider it an extensively useable security measure in its own right. DL has a long way to go before it is mature enough to be deployed as a standalone Unified Threat Management (UTM) environment. In this paper, we have discussed the aspects an organisation should keep in mind when deploying a DL based solution. In addition, we have also included a list of features that would be useful to security practitioners if they can be provided by the DL base mechanisms.

In conclusion, DL has a lot of promise and with the right features, it could become an impactful tool in the security and privacy arsenal. With the increase of sophistication and complexity of future technology in the current infrastructure, AI-based security and privacy countermeasures (ML/ DL) might be the next logical step. For this reason, cybersecurity researchers have to become active participants in the ML/ DL evolution, rather then just deploying them to security and privacy problems as off-the-shelf kits.

References

1. "Internet security threat report," Symantec Corporation, Annual Report - Online Volume 23, 2018.
2. S. Morgan, "2017 Cybercrime report, cybercrime damages will cost the world us$6 trillion by 2021," Cybersecurity Ventures, Herjavec Group, Online Report, 2017.
3. J. Trull, "Top 5 best practices to automate security operations," Microsoft Secure, Enterprise Cybersecurity Group, Online Blog, August 2017.
4. "2018 Cost of a Data Breach Study: Global Overview," Ponemon Institute - Benchmark research sponsorded by IBM Security, Online Report, July 2018.
5. "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal of the European Union*, vol. L119/59, May 2016. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC
6. S. Ciccone, "Cybersecurity: More threats, but also more opportunities," Paloalot Networks, Online, June 2016.
7. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
8. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
9. W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 conversational speech recognition system," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5934–5938.
10. G. H. Yann LeCun, Yoshua Bengio, "Deep learning," pp. 436–444, 2015.
11. Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-Sec: deep learning in android malware detection," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 371–372.
12. J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *Malicious and Unwanted Software (MALWARE), 2015 10th International Conference on*. IEEE, 2015, pp. 11–20.
13. M. Armstrong, "The future of a.i.." Statista Infographics, Statista, Online Report, November 2016.
14. N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.
15. N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *25th USENIX Security Symposium (USENIX Security 16), Austin, TX*, 2016.
16. Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-Sec: Deep learning in Android malware detection," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 371–372, Aug. 2014. [Online]. Available: http://doi.acm.org/10.1145/2740070.2631434
17. B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *AI 2016: Advances in Artificial Intelligence*, B. H. Kang and Q. Bai, Eds. Cham: Springer International Publishing, 2016, pp. 137–149.

18. S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.
19. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 21–26.
20. R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 305–316.
21. M. Zolotukhin, T. Hämäläinen, T. Kokkonen, and J. Siltanen, "Increasing web service availability by detecting application-layer DDoS attacks in encrypted traffic," in *Telecommunications (ICT), 2016 23rd International Conference on*. IEEE, 2016, pp. 1–6.
22. X. Yuan, C. Li, and X. Li, "Deepdefense: Identifying DDoS attack via deep learning," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2017, pp. 1–8.
23. M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.
24. P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.
25. R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 2015, pp. 1310–1321.
26. J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "iPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1005–1016, 2017.
27. Z. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.
28. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspective," pp. 1798–1828, 2013.
29. Y. Bengio, "Deep learning of representations: Looking forward," in *International Conference on Statistical Language and Speech Processing*. Springer, 2013, pp. 1–37.
30. ——, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, ser. Proceedings of Machine Learning Research, vol. 27. PMLR, 2012, pp. 17–36.
31. X. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," pp. 514–525, 2014.
32. Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.
33. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.
34. N. Srivastava and R. Salakhutdinov, "Multimodal learning with Deep Boltzmann Machines," pp. 2949–2980, 2014.

35. J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.
36. I. Arel, D. C. Rose, T. P. Karnowski *et al.*, "Deep machine learning- A new frontier in artificial intelligence research," pp. 13–18, 2010.
37. I. S. Rafal Józefowicz, Wojciech Zaremba, "An empirical exploration of recurrent network architectures," in *ICML*, 2015.
38. D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *International Conference on Machine Learning*, 2015, pp. 2113–2122.
39. Y. B. James Bergstra, "Random search for hyper-parameter optimization," JMLR, pp. 281–305, 2012.
40. M. Denil, B. Shakibi, L. Dinh, N. De Freitas *et al.*, "Predicting parameters in deep learning," pp. 2148–2156, 2013.
41. S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.
42. L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998.
43. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *31st Conference on Neural Information Processing Systems (NIPS)*, 2017.
44. G. Zhong, L.-N. Wang, X. Ling, and J. Dong, "An overview on data representation learning: From traditional feature learning to recent deep learning," *The Journal of Finance and Data Science*, vol. 2, no. 4, pp. 265–278, 2016.
45. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," ACM, pp. 675–678, 2014.
46. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, March 2016, pp. 372–387.