Energy-Based Learning for Preventing Backdoor Attack

Xiangyu Gao[1] and Meikang Qiu[2(✉)]

[1] New York University, New York City, NY, USA
xg673@nyu.edu
[2] Texas A&M University Commerce, Commerce, TX, USA
qiumeikang@yahoo.com

Abstract. The popularity of machine learning has motivated the idea of *Energy-Based Learning* (EBL), which used *Energy-Based Models* (EBMs) proposed by Prof. Yann to capture dependencies between variables. In addition, the application of several machine learning tools into the field of backdoor becomes widespread as well. However, the current backdoor researches didn't consider the novel EBL tools. This paper studies both EBL methods and backdoor attack of machine learning. We propose an algorithm to leverage energy-based learning for preventing backdoor attack. Several case analysis in this paper has demonstrated the promising of applying energy-based learning to improve the backdoor protection techniques.

Keywords: Energy-based learning · Backdoor · Cyber security · Machine learning · Big data

1 Introduction

With the development science and technology, machine learning has become one of the most important tools in a lot of fields such as cyber security [11], computer vision [47] and high-frequency trading [10]. The ultimate goal for any statistical modeling machine learning model is to precisely capture the dependencies between different encoding variables so that it can be used to do prediction given the value of known variables. Among all potential models, *Energy-Based Models* (EBMs) [20] are popularly used. To be specific, EBMs will try to achieve the dependencies between variables by associating a scalar energy to each configurations. Afterwards, inference then participates in setting values of observed variables and then finds values of the remaining variables that minimize the energy.

Compared with other learning process, *energy-based learning* (EBL) proposed by Yann [19] provides a unified framework for many probabilistic and non-probabilistic approaches to learning. In other words, it can be considered as an alternative to probabilistic estimation for different learning tasks such as classification [16] and decision-making [50]. In addition, the fact that energy-based learning does not have any requirements for proper normalization [1] brings itself

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13370, pp. 706–721, 2022.
https://doi.org/10.1007/978-3-031-10989-8_56

many benefits. First of all, it can naturally avoid problems related to estimate the normalization constant in probabilistic models. In addition, this brings a lot of flexibility in the design of learning machines.

Given the uniqueness of EBL, we believe it a good fit for the backdoor field. In cyber security world, backdoor [3] refers to any method by which authorized and unauthorized users are able to get around the normal security system and gain high level user access on a computer system. Once they are in, cyber criminals can use backdoor to steal valuable data [36,41] or install hijack devices [48].

Due to the development of machine learning, people start to rely on the third-party platforms (e.g. AWS [28], Azure [6] or Google Cloud [17]) to train their data. In addition, machine learning users also prefer to store their data on the third-party so as to avoid too much disk occupancy in their local machine. In spite of the convenience, these are all based on the assumption that the third-party platform are trustworthy. However, a series of malicious attacks [9] have proved that the full trust to third-party platforms is dangerous. Specifically, if the adversary injects a backdoor into the training set, the machine learning results might be misleading since the attackers will "instruct" the model to output unrealistic results they want. To make things worse, usually it is quite hard for the developers to find these backdoor attack since only a tiny change of the samples is sufficient to generate the misleading result.

In response to the above concerns, in this paper, we first give an overview of several achievements of both energy-based learning and backdoor attack. Then for EBL, it is important to shows its working mechanism and highlights the advantages over other methods. As for the backdoor attack, there is no bias on either the current research in attacker's strategy or defender's strategy. The goal is to figure out the potential to transplant the energy-based learning into the field of backdoor. There are three main contributions of this paper:

- An systematic study of the current trend in energy-based learning and backdoor methods.
- A proposed algorithm to leverage energy-based learning to enhance the prevention of backdoor attack.
- A case study to figure out the potential benefits of our proposed algorithm.

The remainder of this paper is organized as follows. Section 2 summarizes the features and the current trend of energy-based learning, including more descriptions for several loss functions. Then, Sect. 3 studies recent techniques in backdoor. It will present the techniques on both attackers' and defenders' side. Furthermore, Sect. 4 gives a detailed description of our proposed algorithm, followed a case study in Sect. 5. Afterwards, Sect. 6 discusses about how to combine energy-based learning together with backdoor by listing several possible directions. Finally, we conclude the paper in Sect. 7.

2 Energy-Based Learning Overview

In this section, we will give an overview of the background of energy-based learning by comparing it against some of the other machine learning techniques.

The goal is to show several research trends of this field so as to discover more potential application situations.

2.1 Energy-Based Models Overview

The rapid development of machine learning has already shown its usefulness in many fields, such as finance [12,40], tele-health [33,39], and transportation [34,35]. The general framework of machine learning [21,32,38] is to output the values of target variables given the values of input data. An important metric to measure the trained model is that how far its output is from the correct result. The energy-based models leverage a reasonable energy function which represents the "goodness" (or "badness") of each possible configuration X and Y.

There are several scenarios where the EBM can be considered as a good fit. The first one is prediction and classification. This situation is quite visualized in Fig. 1 where given the input X, we want to get the value of Y that is most compatible with the current input. The second one is about the ranking, which is more complex than the first scenario. Specifically, we can consider the prediction and classification as finding the minimum value among all candidates while regard ranking as a sorting problem. However, they are similar essentially since the ultimate goal is to compare the energy value among multiple configurations. The third one is about detection, which is quite popular in the field of image recognition. Given the input X and output Y, we want to see whether Y could be the possible result by comparing the energy value of such configuration with a specific threshold. The last scenario, conditional density estimation, usually occurs when the output Y is only an intermediate output that is fed to the input of another, separately built system.

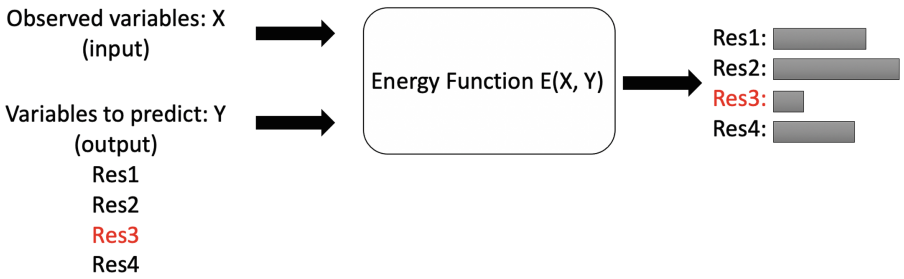

Fig. 1. Energy function E(X, Y) is used to measure the compatibility between the input X and the output Y. The model selects the output that minimizes the energy E.

Among all these four scenarios mentioned above, the core of EBMs is the equation below.

$$Y^* = argminE(X, Y) \tag{1}$$

It uses the convention that the highly compatible configurations of variable have small energy values while highly incompatible configurations of the variables

means large energy value. Therefore, in the energy-based learning framework, we want to select reasonable energy functions so that it can clearly differentiate compatible configurations from incompatible configurations easily.

2.2 Loss Function in Energy-Based Learning

The ultimate goal for energy-based learning is to output a configuration between the input variables X and the output variables Y for new input samples that do not belong to the training data set. However, only the data within the training set is visible, what we could do is to divide them into both training set and testing set so that our trained model can be guaranteed to have good performance in the testing set. In order to achieve this goal, we need to develop specific metrics measuring the distance between the correct results and the current output, which can be regarded as the loss function selection. A good loss function [5] can clearly differentiate the correct solutions from the wrong ones. We would overview several typically used loss function below.

The first and most simplest one is called the energy loss defined as

$$L(Y^i, E(W, Y, X^i)) = E(W, Y, X^i) \tag{2}$$

This is one of the most commonly used in situations such as regression [42] and neural network [52]. However, when applying this loss function into other architectures, it cannot output the desired answer because the loss fails to pull up on any other energy even if it can push down the energy of the desired answer. In other words, the energy loss will only work with architectures that can work in a way to push down on $E(W, Y^i, X^i)$ together with pull up energies of the other answers.

The second one is called generalized perception loss defined as

$$L(Y^i, E(W, Y, X^i)) = E(W, Y^i, X^i) - minE(W, Y, X^i) \tag{3}$$

If we take a close look at the loss function, we can find that its value is always non-negative since the second term is the lower bound of the first term. In addition, this loss function provides some room to push down on the energy of the desired answer while pulling up on the energy of other answers. To be specific, when we increase the value of $E(W, Y^i, X^i)$ when Y^i is not the desired answer, the first term will increase while the second term will keep the same. The perception loss has been used in settings such as handwriting recognition [31] and speech tagging [44]. However, its major deficiency is that there is lack of mechanisms to enlarge the gap between correct answer and incorrect ones, making it harder for people to differentiate them.

The third one is called generalized margin loss defined as

$$L(W, Y^i, X^i) = Q_m(E(W, Y^i, X^i), E(W, \bar{Y}^i, X^i)) \tag{4}$$

where $Q_m(x_1, x_2)$ is a convex function whose gradient has a positive dot product with vector $(1, -1)$ in the region where $E(W, Y^i, X^i)+m > E(W, \bar{Y}^i, X^i)$ Margin

losses can represent several loss functions such as the hinge loss [13], log loss [51], minimum classification error loss [15], and square-exponential loss. Some form of margin should be created to enlarge the gap between the correct answers and the incorrect ones.

2.3 Features and Application of EBM Frameworks

The energy-based learning offers a unified framework for many probabilistic and non-probabilistic approaches to learning because it can be considered as an alternative to estimation for prediction, classification, or decision-making task. In addition, as for energy-based learning, there is no requirement for normalization, which is necessary for probabilistic models. Therefore, this gives EBM framework much more flexibility in the design of learning machines. In fact, many existing learning models can be expressed simply in the framework of energy-based learning.

3 Techniques in Backdoor

In this section, we want to make an overview of the backdoor techniques [23]. It will start from a general description of the backdoor concept followed by current backdoor trends, including the development on both the attackers side and the defenders side. We are trying to add some insights for potential improvements for better attacking and defending strategies.

3.1 Backdoor Concept

Traditional backdoor refers to a malicious code piece embedded by an attacker into one system so that the attacker can obtain higher privilege than allowed. For example, the attacker can circumvent the authentication system by inputting his/her own password. Usually, such action is hard to detect because the whole system works normally most of the time except when the attacker is asked to input a password. Y. Zeng and M. Qiu et al. had proposed several novel algorithms to prevent backdoor attack, such as clean label techniques [56] and DeepSweep [37].

Given the popularity of backdoor attack, people even start to consider the triggers [54,57] of the backdoor attack. In general, the conclusion is that if there is inconsistency between the attack for test cases and the attack for training cases, the attack would be vulnerable [24].

3.2 Attackers in Backdoor

Usually, the attackers can implement their backdoor attack in three aspects: data set [3], platform [4] and model [18]. To make things worse, these three parts are not orthogonal to each other. An attacker can put the attack in all three aspects at the same time.

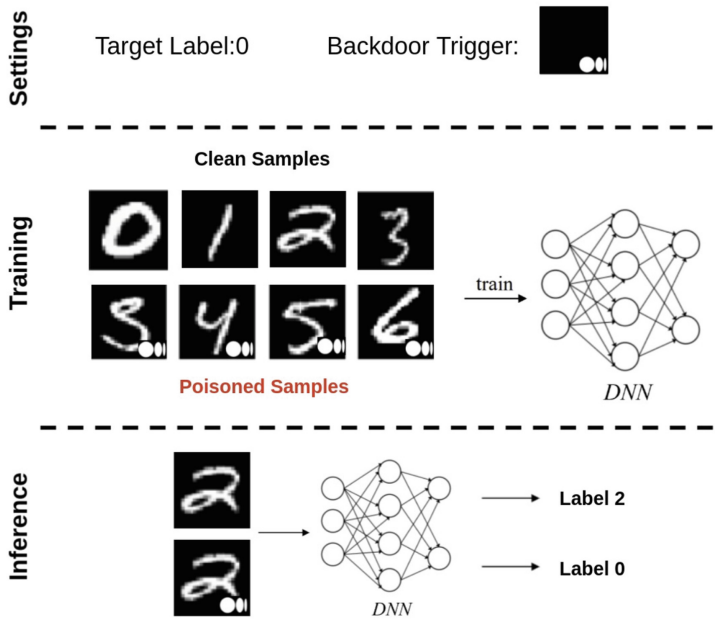

Fig. 2. An example to show how backdoor attacks work.

The backdoor attack to data set is illustrated in Fig. 2. Due to the privilege obtained by the attacker, he/she have the access to update the data set. Therefore, the attacker adds some poisoned samples which have some white circles in the bottom right and labels them as zero. Then the poisoned data set will be fed into models such as deep neural network to generate the inference model. The poisoned samples will add a lot of redundant misleading features which will decrease the accuracy of the model output. For example, in Fig. 2, for any input with white circles in the bottom right, the trained deep neural network will label it as zero rather than consider other pixels.

As for the backdoor attack for the platform, after the users provide their data set. Model structure, and the training schedule to one third-party platform, the attackers will try to modify the training process. For instance, the attacker instructs the platform to train the model before cleaning the data [43], which is a necessary for machine learning. If this happens, it will lead to the output result far from correctness. In other words, even if the attacker cannot modify anything inside the data, changing the training process can generate bad result as well.

Also, the backdoor attack for the model is another venue for the attackers to use so as to influence the output result. Specifically, if the users want to train the data set on deep neural network through third-party platform, they need to provide the model structure beforehand. However, after receiving the model, the attacker can modify the model into a new one, by removing some layers or

removing some computation nodes in DNNs, so that the final output result will be subjective to what the attacker wants.

3.3 Defenders in Backdoor

In response to a series of backdoor attack behaviors, defenders take some corresponding action to detect attacking clues and avoid the side effect from the attackers. The ultimate goal for the defenders is to train their data on the model they want by following the process determined by themselves. In other words, they want to manipulate the data set to avoid the backdoor attack.

In this scope, Qiu et al. used Deepsweep [37] to investigate the effectiveness of data augmentation techniques to mitigate backdoor attacks and enhance DL models' robustness; Datta et al. [8] draws a conclusion that the increasing number of independent attackers will even reduce the possibility of successful backdoor attack, which means there might be some internal friction among all attackers; knowledge distillation [55] is leveraged in terms of removing poison data from a poison training data set and recovering the accuracy of the distillation model.

4 Our Approach

Based on the description of energy-based learning techniques and backdoor methods, in this section, we want to propose our approach to combine these two together. Specifically, we want to explore the possibility to put energy function into the constraints part rather than the objective function. Then, we can try to solve an optimization problem with customized objective function, or find a feasible solution.

4.1 Energy Function as a Constraint

As for most of the points in the discussion above, we consider the energy function as the objective one and try to find a suitable candidate to measure our goal. In fact, we can reallocate the energy function into other places. For instance, we could set up some customized objective function with the energy function as one constraint. Therefore, in this part, rethinking the role of energy function is our main topic.

To be specific, all resources such as computation resources and storage memory are limited. Hence, in a lot of situations, what the users care most about is a reasonably good choice or an acceptable sub-optimal result. In response to this demand, we believe that we should put the energy function in the constraint part by choosing one energy function and restricting its value to be less than or equal to a predefined threshold. The choice of objective functions [46] can be prone to the users' preference. The concrete format can be shows as following:

$$\begin{aligned} \underset{X,\alpha}{\text{minimize}} \quad & f(X,\alpha) + \text{regularization function} \\ \text{subject to} \quad & E_i(X,Y) \leq c_i,\ i = 1,\ldots,m. \\ & X \in \text{Input Range} \\ & \alpha \in \text{Parameter Range} \\ & \text{Other Constraints} \end{aligned}$$

The objective function can be determined by the users depending on their preference. In addition to other normal constraints (e.g., the range of parameters), we add new constraints by setting the energy functions to be within a threshold. Typically the smaller the energy function, the better the effect. Therefore, these constraints can force the output result to be reasonably good within a range. Solving these optimization problems might involve convex and nonconvex solvers selection [14] or new solvers development. These developments are orthogonal to our approach.

In addition to solve the optimization problem with selected objective functions, it is also possible for us to solve a satisfiable problem by leveraging practical solvers [25]. Switching from optimization problem to satisfiable problem is quite simply. We can stick to the optimization solver with the objective function f to be independent of all the constraints. Therefore, whatever the constraints' function format, they will not affect the final optimal value of the objective function. The goal for this problem is to find a feasible solution which satisfies all the constraints. By comparing against several solvers [29,49], we can finally find the most efficient and effective one for our framework.

4.2 Algorithm to Implement Our Approach

According to our description of the proposed method, in this subsection, we are trying to present an algorithm showing how to implement our approach in more details.

In general, this proposed algorithm switch the position of customized objective function and the current energy loss function. Usually, the users will have some threshold in mind before making the optimization problem. For instance, in linear regression, if the value of R^2 or adjusted R^2 [27] is too small, people need to switch to other candidate models. Therefore, our algorithm provides us with chances to give threshold beforehand (which might not need to be optimal) but offers another opportunity to give more customized objective functions. Then the algorithm collects information including input data, current constraints, customized objective functions and a list of threshold representing the users' expectation of the final value of their objective function. The algorithm also picks up one solver for this optimization problem. Then, the algorithm will regard the customized function as the objective function. It will go through all threshold within the list and try to solve a feasible solution and jump out of the loop whenever there is a solution. The output includes the threshold value of the energy function and final optimal output value.

Algorithm 1. ***Backdoor Prevention*** Algorithm

Require: N input pairs, M constraints, the customized objective function P, estimated threshold list $[c_1, c_2, ..., c_k]$ for the objective function, and the current objective loss function F.

Ensure: An updated optimization problem to prevent the backdoor attack.

Force the threshold list to be sorted in increasing order
Replace the current objective loss function F by customized objective function P
Remain the current M constraints
Add new constraint which represent the users' anticipation of the energy function. It gives the upper bound of the energy function. The format should be $F(X, Y) \leq c_i$

```
Choose one solver to solve this optimization function.
while (1)
    set the threhold of energy function to be c_i
    solve the optimization problem
    if (there is a solution)
        break;
    else
        i++;
```

Output results: The final results include the threshold value of the energy function and the optimal value of the customized objective function.

5 Case Study

In this section, we are going to present some benefits of our proposed algorithm by showing potential application in several fields, such as monitoring the attack in iPhone and web mail server. Then, two scenarios, equilibrium between attackers and defenders together with the energy function selection, are mentioned to highlight more benefits.

5.1 Backdoor Attack in iPhone

In 2016, there is a debate between Apple vs the U.S. government in terms of iPhone [7,53]. On the one side, Apple together with digital rights groups advocating protection of customer digital privacy want to protect the privacy of all users; on the other side, the U.S. government and the FBI wants to force the company to unlock data to provide crucial evidence which could be helpful to detect attack from terrorists. To make thing worse, this debate will last forever since there is no absolutely correct answer to both sides.

In order to solve this problem, traditional machine learning goal to prevent backdoor attack might be out-of-date because it is hard to create one loss function to measure the debate from both sides. However, our proposed energy-based learning can be quite helpful in this part. Specifically, we can leverage two energy functions, each of which measures the requirement from either the privacy advocates and the information requestors. After putting them into the constraints

of the optimization problem, it is possible for us to get an optimal or a feasible solution based on the users' requirement.

5.2 Backdoor Attack in Web Mail Server

Web mail server is another place where a lot of backdoor attack might happen. Given the situation that a lot of spam detection methods have been implemented in mail server, the attackers have sought to bypass these attack detectors. For instance, they might use some strange ways to express their sensitive contents so that the detectors cannot label these mails as spam message. Although the ultimate goal of the system developers is to rule out all spam emails, the limited cost to implement their attack detection system cannot allow them to achieve this ideal goal.

Therefore, it is necessary to leverage our energy-based learning in the area. Rather than do a classification problem which tries its best to differentiate spam email from normal ones, we put the misclassification rate into the constraint part and bound it by one threshold. In addition, the cost to main the system can also be regarded as a constraint. Then solving one feasible solution within these constraints can output an acceptable solution to us.

5.3 Equilibrium Among the Arm-Race Between Backdoor Attackers and Defenders

We believe that the relationship between attackers and defenders will exist for a long time. There is always an arm race [58] between attackers and defenders in the field of backdoor. In order to avoid the forever arm race in between, we want to find the equilibrium point in between. Specifically, we can develop two energy functions, one for the attackers and the other one for the defenders. Then, the output of these two energy functions will determine the level of difficulty for each counter-party to implement their action. If both of them are greater than some specific threshold, it means that neither attackers nor the defenders have strong willingness to continue putting any efforts to this system, which can be regarded as an equilibrium.

In order to calculate the equilibrium [22,30] status, the selection of the energy functions for both counter-parties is quite important. These two functions should objectively reflect the reward for attackers and defenders. When we are given a series of energy functions, it is important to train them by feeding current backdoor attacks data. Then, as for newly developed system, the developers are willing to consider the equilibrium status beforehand so that they will try to make the system within the equilibrium range. After that, even if the developed system is still not bug-free, the attackers do not have strong preference to implement any backdoor attack.

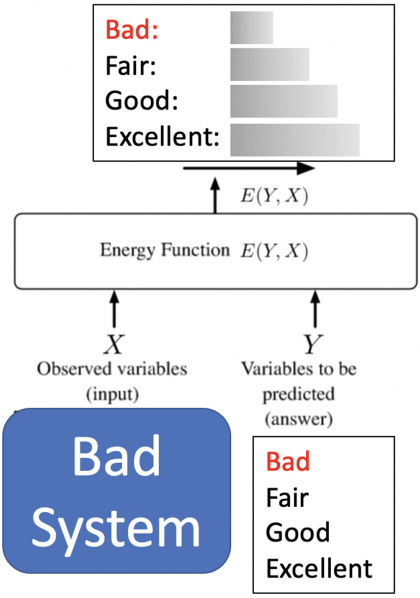

Fig. 3. Energy function E(X, Y) is used to measure the safety level of the input system.

5.4 Energy Function Selection for Safety Level Measurement

As for the energy function selection to measure the safety level of the current system design, typically, both the system developers and the users, even for the attackers, want to quickly understand whether a given system is safe enough to use. Therefore, it would be valuable to design an energy function to measure such safety level.

Naturally, the lower energy value represents the higher safety level of the current system. Based on this consensus, we want to build an energy-based model and train this model among several current available systems in Fig. 3. The input can be considered as the features of those system while the output would be the safety level of them. The goal is to find the value of parameters of the energy function's skeleton. After training within these existing systems, we want to implement the energy function into newly built ones to have an estimation for whether its current status is safe enough to use by comparing its value with a particular threshold. A reasonable choice of the energy function can be helpful to show whether a given system is safe enough to use.

This direction is useful because currently, most of the existing system safety analysis tools [45] involves brute-force testing (e.g. software test [2]). In other words, people need to come up with mechanisms to develop test cases automatically so that it is highly possible that the current scenario is bug-free. This process is quite time-consuming and money consuming. What we really want is a general model which can quickly inference the safety level of the product. At least, it is able to easily rule out the possibility that bad system will go to the market.

6 Discussion

Given the overview of both energy-based learning ideas and backdoor attacks, we believe it would be quite promising to implement the energy-based learning tools into the backdoor field. Therefore, in this section, we want to propose several directions worth further research.

Generally, the discussions are mainly about how easy it would be for the attackers to hijack the system by using one particular strategy, and how hard it would be for the defenders to avoid the attack with any defending tools.

6.1 Energy Function Design to Detect Backdoor Attack Behavior?

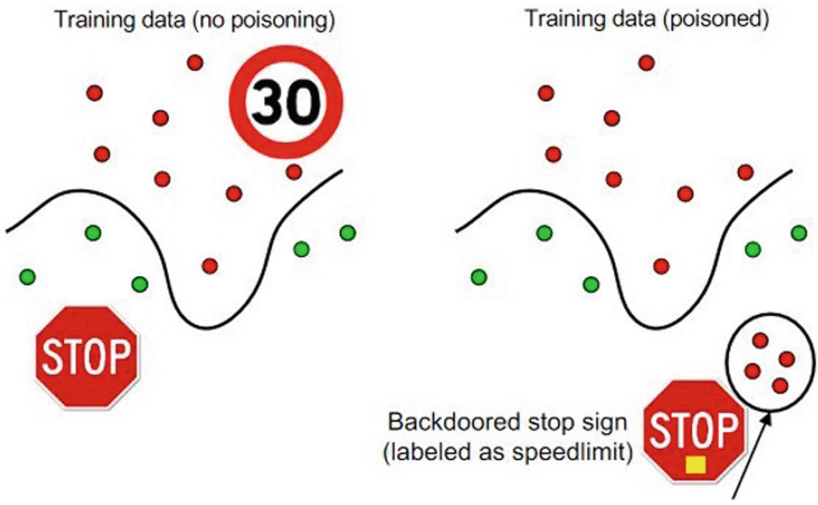

Fig. 4. Energy function E(X, Y) is used to measure the sanity level of the data set.

We also consider the energy function design for backdoor attack detection would be an interesting direction. Just as shown in Fig. 2, sometimes if the untrusted third-party add small amount of poisoned data into the training set similar to Fig. 4, the output result might diverge from what it is supposed to be. Therefore, it is vital to come up with an energy-based model to quickly check whether the existing data is still "clean" and reliable or not. In other words, we hope the energy function can help detect the attack behavior inside the data set quickly.

In general it can be regarded as a classification problem. The input would be the current training data set while the output is a value measuring the reliability level of the current status. Given the fact that people have strong preference to store their data in the third-party platforms and sometimes those platforms might face malicious attack, it is important to do some sanity check [26] before reusing the training data for machine learning tasks. A naive solution might be

storing the data in multiple places and then comparing against them whenever using the data. However, it takes a long time to compare against two super large data set, which makes the machine learning task much longer than necessary. Another possible solution is to randomly select samples from the data set and only check consistency among these samples. However, how many samples we should select and how to select these samples remain a big problem to implement reliable sanity check.

Therefore, if it is possible for us to choose one energy function that could easily detect the change of the data set, the whole process can be finished much more quickly. To be specific, the input of the energy function would be two data set and the energy function itself will randomly select some of the critical parts within the data set. If there is no big difference in between, the absolute difference between the output value of the energy function in two versions of this data set will be smaller than the predefined threshold. How to determine the critical parts of the data set and how to design the corresponding energy function would be an interesting topic.

6.2 Energy Function Design to Implement a Backdoor Attack Efficiently?

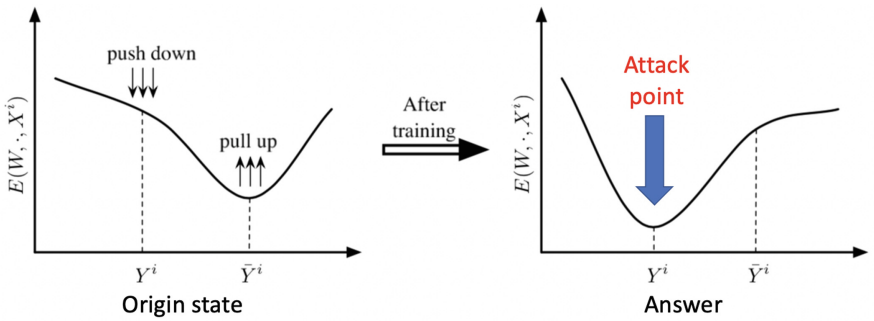

Fig. 5. Energy function E(X, Y) is used to find the best place to implement an attack.

As for the attackers, an energy function is also useful for them to pick up efficient and effective strategy to implement their backdoor attack similar to [3]. When the attackers decide to put some noise into the data set either by changing the existing ones or adding misleading data, they also want to hide their action from the defenders. Therefore, a good strategy is helpful to come up with in order to avoid the detection from the defender. There might be several dimensions to consider. For example, how much of the existing data should be modified? The less the better; how close the updated data set is compared with the original one? The closer the better; how easy it is to implement this modification to the data set? The easier the better.

We can consider this question as a ranking problem. The input to the energy function includes the features of existing system and a series of attack strategies. Then, the energy function will output values which represent the level of difficulty to implement those strategies in Fig. 5. It will rank them in some specific order and then provide more useful feedback to the attackers. The attackers can combine the ranking result from the predefined energy function with the potential cost to each strategy, before taking the action.

7 Conclusion

In this paper, based on the study of current development of both energy-based learning and backdoor, we proposed some novel approaches to leverage the energy-based learning tools for preventing backdoor attack to machine learning. The case studies in this paper had demonstrated the effectiveness of using energy-based learning to prevent backdoor attack, which is a critical threat to machine learning. The promising usage of energy-based learning will greatly impact the security aspect of machine learning.

References

1. Ali, P., Faraj, R., Koya, E., Ali, J., Faraj, R.: Data normalization and standardization: a technical report. Mach Learn Tech report (2014)
2. Beizer, B.: Software system testing and quality assurance. Van Nostrand Reinhold Co. (1984)
3. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
4. Chen, X., Ma, Y., Lu, S.: Use procedural noise to achieve backdoor attack. IEEE Access. **9**, 120204–120216 (2021)
5. Christoffersen, P., Jacobs, K.: The importance of the loss function in option valuation. J. Fin. Econ. **72**, 291–318 (2004)
6. Copeland, M., Soh, J., Puca, A., Manning, M., Gollob, D.: Microsoft Azure. Apress, New York (2015)
7. Coutros, G.: The implications of creating an iPhone backdoor. Natl Sec. L. Brief **6**, 81 (2016)
8. Datta, S., Lovisotto, G., Martinovic, I., Shadbolt, N.: Widen the backdoor to let more attackers in. arXiv preprint arXiv:2110.04571 (2021)
9. Delac, G., Silic, M., Krolo, J.: Emerging security threats for mobile platforms. In: 34th International Convention MIPRO (2011)
10. Fang, B., Feng, Y.: Design of high-frequency trading algorithm based on machine learning. arXiv preprint arXiv:1912.10343 (2019)
11. Ford, V., Siraj, A.: Applications of machine learning in cyber security. In: 27th International Conference on Computer Application in Industry and Engineering (2014)
12. Gai, K., et al.: Efficiency-aware workload optimizations of heterogeneous cloud computing for capacity planning in financial industry. In: IEEE CSCloud (2015)
13. Gentile, C., Warmuth, M.: Linear hinge loss and average margin. In: Advances in Neural Information Processing Systems (1998)

14. Hiriart-Urruty, J.: From convex optimization to nonconvex optimization. Necessary and sufficient conditions for global optimality. In: Clarke, F.H., Demyanov, V.F., Giannessi, F. (eds) Nonsmooth Optimization and Related Topics. Ettore Majorana International Science Series, vol. 43, pp. 219–239. Springer, Boston (1989). https://doi.org/10.1007/978-1-4757-6019-4_13
15. Juang, B., Hou, W., Lee, C.: Minimum classification error rate methods for speech recognition. IEEE Trans. Speech Audio proc. **53**, 257–265 (1997)
16. Kotsiantis, S., Zaharakis, I., Pintelas, P.: Machine learning: a review of classification and combining techniques. Artif. Intel. Rev. **26**, 159–190 (2006)
17. Krishnan, S., Gonzalez, J.: Building your next big thing with google cloud platform: a guide for developers and enterprise architects (2015)
18. Kwon, H., Yoon, H., Park, K.: Multi-targeted backdoor: identifying backdoor attack for multiple deep neural networks. IEICE Trans. Inf. Sys. **103**, 883–887 (2020)
19. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predict. Struct. Data. **1**, 1–59 (2006)
20. LeCun, Y., Huang, F.: Loss functions for discriminative training of energy-based models. In: International Workshop on Artificial Intelligence and Statistics (2005)
21. Li, Y., et al.: Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. IEEE TII **17**(4), 2833–2841 (2020)
22. Li, Y., Tan, S., Deng, Y., Wu, J.: Attacker-defender game from a network science perspective. Chaos: Interdiscip. J. Nonlinear Sci. 28, 051102 (2018)
23. Li, Y., Wu, B., Jiang, Y., Li, Z., Xia, S.: Backdoor learning: a survey. arXiv preprint arXiv:2007.08745 (2020)
24. Li, Y., Wu, B., Li, L, He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers. In: IEEE/CVF CV (2021)
25. Lin, Y., Schrage, L.: The global solver in the LINDO API. Optim. Methods Softw. **24**(4–5), 657–668 (2009)
26. Lu, H., Xu, H., Liu, N, Zhou, Y., Wang, X.: Data sanity check for deep learning systems via learnt assertions. arXiv preprint arXiv:1909.03835 (2019)
27. Miles, J.: R-squared, adjusted R-squared. Encyclopedia of statistics in behavioral science (2005)
28. Mukherjee, S.: Benefits of AWS in modern cloud. SSRN 3415956 (2019)
29. Müller, T.: ITC 2007 solver description: a hybrid approach. Ann. Oper. Res. **172**, 429–446 (2009)
30. Pirani, M., Nekouei, E., Sandberg, H., Johansson, K.: A graph-theoretic equilibrium analysis of attacker-defender game on consensus dynamics under h_2 performance metric. IEEE TNSE, pp. 1991–2000 (2020)
31. Plamondon, R., Srihari, S.: Online and off-line handwriting recognition: a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 63–84 (2000)
32. Qiu, H., Dong, T., et al.: Adversarial attacks against network intrusion detection in IoT systems. IEEE IoT J. **8**(13), 10327–10335 (2020)
33. Qiu, H., et al.: Secure health data sharing for medical cyber-physical systems for the healthcare 4.0. IEEE JBHI **24**, 2499–2505 (2020)
34. Qiu, H., et al.: Topological graph convolutional network-based urban traffic flow and density prediction. IEEE TITS. **22**, 4560–4569 (2020)
35. Qiu, H., Qiu, M., Lu, R.: Secure V2X communication network based on intelligent PKI and edge computing. IEEE Netw. **34**(42), 172–178 (2019)
36. Qiu, H., Qiu, M., Lu, Z.: Selective encryption on ECG data in body sensor network based on supervised machine learning. Info. Fusion **55**, 59–67 (2020)

37. Qiu, H, Zeng, Y., Guo, S., Zhang, T., Qiu, M., Thuraisingham, B.: Deepsweep: an evaluation framework for mitigating DNN backdoor attacks using data augmentation. In: 2021 ACM Asia CCS (2021)
38. Qiu, H., Zheng, Q., et al.: Deep residual learning-based enhanced JPEG compression in the internet of things. IEEE TII **17**(3), 2124–2133 (2020)
39. Qiu, L., Gai, K., Qiu, M: Optimal big data sharing approach for tele-health in cloud computing. In: IEEE SmartCloud, pp. 184–189 (2016)
40. Qiu, M., et al.: Data transfer minimization for financial derivative pricing using Monte Carlo simulation with GPU in 5G. JCS **29**(16), 2364–2374 (2016)
41. Qiu, M., Gai, K., Xiong, Z.: Privacy-preserving wireless communications using bipartite matching in social big data. FGCS **87**, 772–781 (2018)
42. William, J., Freund, R., Sa, P.: Regression Analysis. Elsevier, Amsterdam (2006)
43. Rahm, E., Do, H.: Data cleaning: problems and current approaches. IEEE Data Eng. Bull. **23**, 3–13 (2000)
44. Ratnaparkhi, A., et al.: A maximum entropy model for part-of-speech tagging. In EMNLP **1**, 133–142 (1996)
45. Rouvroye, J., Van Den Bliek, E.: Comparing safety analysis techniques. Reliabil. Eng. Syst. Safety **75**, 289–294 (2002)
46. Roy, B.: Problems and methods with multiple objective functions. Math. Progr. **1**(1), 239–266 (1971)
47. Sebe, N., Cohen, I., Garg, A., Huang, T.: Machine Learning in Computer Vision. Springer, Dordrecht (2005). https://doi.org/10.1007/1-4020-3275-7
48. Shao, Z., Xue, C., et al.: Security protection and checking for embedded system integration against buffer overflow attacks via hardware/software. IEEE TC **55**(4), 443–453 (2006)
49. Stellato, B., et al.: Embedded mixed-integer quadratic optimization using the OSQP solver. In: IEEE ECC, pp. 1536–1541 (2018)
50. Tulabandhula, T., Rudin, C.: On combining machine learning with decision making. Mach. Learn. **97**, 33–64 (2014)
51. Vovk, V.: The fundamental nature of the log loss function. In: Beklemishev, L.D., Blass, A., Dershowitz, N., Finkbeiner, B., Schulte, W. (eds.) Fields of Logic and Computation II. LNCS, vol. 9300, pp. 307–318. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23534-9_20
52. Wang, S.: Artificial neural network. In: Interdisciplinary Computing in Java Programming (2003)
53. Wolfson, B., Levy, L.: Impenetrable: Should apple backdoor the iPhone? (2020)
54. Yang, W., Lin, Y., Li, P., Zhou, J., Sun, X.: Rethinking stealthiness of backdoor attack against NLP models. In: 59th Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on NPL (2021)
55. Yoshida, K., Fujino, T.: Disabling backdoor and identifying poison data by using knowledge distillation in backdoor attacks on deep neural networks. In: 13th ACM Workshop on Artificial Intelligence and Security (2020)
56. Zeng, Y, Pan, M., Just, H., Lyu, L., Qiu, M., Jia, R.: Narcissus: a practical clean-label backdoor attack with limited information. In: CoRR abs/2204.05255 (2022)
57. Zeng, Y., Park, W., Mao, Z., Jia, R.: Rethinking the backdoor attacks' triggers: a frequency perspective. In: EEE/CVF CV (2021)
58. Zhang, Y., Paxson, V.: Detecting backdoors. In: USENIX Security Symposium (2000)