Security Analysis of Machine Learning Systems for the Financial Sector

Shiori Inoue and Masashi Une

The use of artificial intelligence, particularly machine learning (ML), is being extensively discussed in the financial sector. Information technology (IT) systems using ML (ML systems), however, tend to have specific vulnerabilities as well as those common to all IT systems. To effectively deploy secure ML systems, it is critical to consider in advance how to address potential attacks targeting the vulnerabilities. In this paper, we classify ML systems into twelve types on the basis of the relationships among entities involved in the system and discuss the vulnerabilities and threats, as well as the corresponding countermeasures for each type. We then focus on typical use cases of ML systems in the financial sector, and discuss possible attacks and security measures.

Keywords: Artificial intelligence; Machine learning system; Security; Threat; Vulnerability
JEL Classification: L86, L96, Z00

Shiori Inoue: Associate Director, Institute for Monetary and Economic Studies (currently, Financial Markets Department), Bank of Japan (E-mail: shiori.inoue@boj.or.jp)
Masashi Une: Director, Institute for Monetary and Economic Studies, Bank of Japan (E-mail: masashi.une@boj.or.jp)

The authors would like to thank Jun Sakuma (the University of Tsukuba) for useful comments. The views expressed in this paper are those of the authors and do not necessarily reflect the official views of the Bank of Japan.

I. Introduction

The use of artificial intelligence (AI) is now being actively discussed in various fields including the financial sector. Financial institutions are considering the use of AI for the provision of such services as deposit-taking, lending, asset management, and insurance brokerage. For instance, greeting clients using a chatbot, forecasting market trends for making recommendations to clients, and estimating a borrower's future earnings for extending loans are regarded as fruitful applications of AI to improve productivity and business risk management. When deploying a new information technology (IT) such as AI, it is necessary to pay close attention to relevant security risks.

In general, AI refers to the use of computer systems and various techniques to perform human-like intelligent activities such as inference, recognition, and decision making. It is implemented by using machine learning (ML). Various studies have identified the vulnerabilities and threats specific to IT systems implemented using ML (Une [2019], Yoshioka [2018]).[1] Manipulation of such vulnerabilities causes critical security incidents such as the theft or alteration of the ML model and/or its training data. To effectively deploy secure ML systems, it is necessary to consider in advance countermeasures against possible adversarial attacks.

In this paper, we classify ML systems into twelve types on the basis of the relationships among entities involved in the system, and discuss the vulnerabilities and threats as well as the corresponding countermeasures for each type. The analysis presented is useful to financial institutions when they are considering which security measures to implement because it enables them to identify possible attacks on and the corresponding countermeasures for the type of ML system to be deployed.

In section 2, we provide an overview of ML systems and describe the method used to classify ML systems. In section 3, we discuss the attacks on and security measures for each type of ML system. In section 4, we focus on typical use cases of ML systems in the financial sector and discuss possible attacks and security measures.

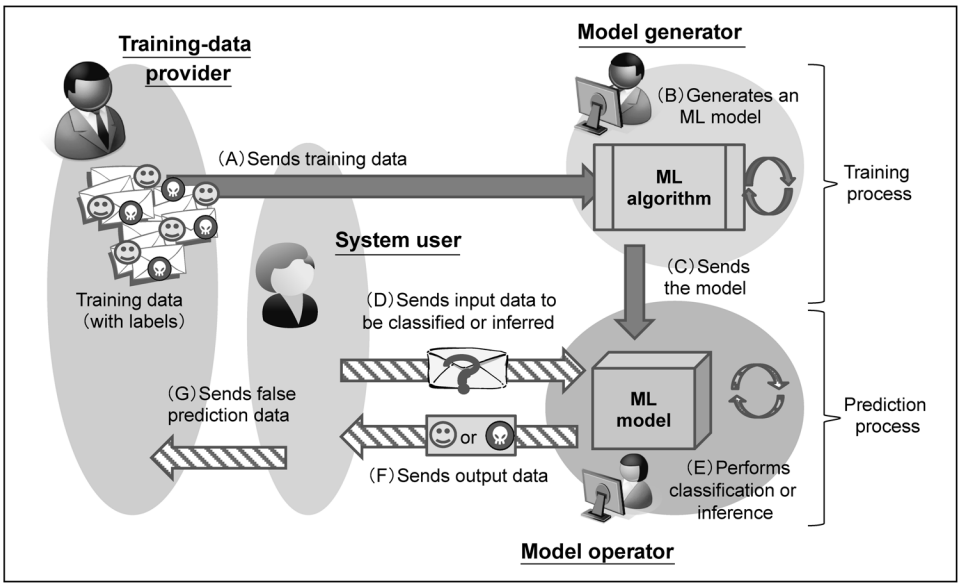
II. Overview and Classification of Machine Learning Systems

A. Entities and Processes

An ML system is generally constructed with four entities: a model generator, a model operator, a system user, and a training-data provider (Une [2019]).[2] The model generator generates an ML model by inputting training data into an ML algorithm. The model operator uses the generated model to perform classification or inference. The system user requests the model generator to create an ML model and/or requests the model operator to perform classification or inference with the model. The training-data provider prepares and provides a set of training data to the model generator. The training process comprises steps (A) to (C) below, and the prediction process comprises steps (D) to (G) (see Figure 1).

1. We call an IT system implemented using ML a "machine learning system" or an "ML system."
2. The focus here is on ML systems using supervised learning, for which the training data include feature values and the corresponding labels.

Figure 1 Training and Prediction Processes in Machine Learning Systems



(A) The training-data provider collects raw data, extracts the feature values of the raw data, assigns a label for each of the feature values in cooperation with the system user, and sends pairs of feature values and the corresponding labels to the model generator as training data.[3]

(B) The model generator generates an ML model by inputting the training data into a predetermined ML algorithm.

(C) The model generator sends the model to the model operator.

(D) The system user sends input data to be classified or inferred to the model operator.

(E) The model operator performs the classification or inference by inputting the data into the model.

(F) The model operator sends the output data to the system user.

(G) In some cases, the system user sends pairs of input and output data to the training-data provider. For instance, when false classification or false inference occurs, the system user sends the training-data provider the pair of input and incorrect output data as well as the correct output data for use in improving the model. The data sent are called "false prediction data."

B. Classification of ML Systems

We observe many variations of ML systems in terms of the relationships among the four entities. For instance, a financial institution uses its internal data as training data, generates an ML model, and operates it. Such a system is classified as one in which a single organization acts as all four entities. Alternatively, a financial institution acting as a system user uses an "ML as-a-service (MLaaS)" provided by a cloud vendor acting

3. When the training data include confidential information, masking or encryption of the data is required. For simplicity, such treatment is assumed to be done before sending the data to the model generator.

Table 1 Twelve Types of Machine Learning Systems

Type	Organizations acting as corresponding entities				Number of organizations
	Training-data provider	System user	Model generator	Model operator	
1					4
2					3
3					
4					
5					
6					
7					2
8					
9					
10					
11					
12					1

Note: Cells with same pattern represent same organization in each type.

as both a model generator and a model operator.[4] Such a system is classified as one in which the organization acting as the system user differs from the one acting as both a model generator and a model operator.

Given these considerations, we logically divide ML systems into fifteen types.[5] However, three of them are not realistically deployable. Consider ML systems in which a single organization acts as both the system user and the model generator while another organization acts as the model operator. The system user, which owns the model, has to communicate with the model operator during the prediction process, which is obviously inefficient. Therefore, we ignore these three types and focus on those shown in Table 1.

III. Attacks on and Security Measures for Machine Learning Systems

A. Security Objective

The security objective for ML systems as well as for general IT systems is to ensure confidentiality, integrity, and availability.[6] For confidentiality, it is necessary to ensure a state in which an unauthorized entity, i.e., an attacker, is not able to access the data and functions of the ML system. For integrity, it is necessary to ensure a state in which the attacker is not able to manipulate the data and functions. For availability, it is necessary to ensure a state in which authorized entities are able to access and operate the ML

4. Amazon Machine Learning, Google Cloud Platform, and Azure Machine Studio are example providers of MLaaS. There are client-facing chatbot services for which ML models are generated by using training data gathered from multiple financial institutions (NTT DATA Corporation [2017]).
5. Phong (2017) proposes an ML system in which an organization acting as the training-data provider participates in the training process. In this case, we consider the model generator to be the entity that finally generates the complete ML model.
6. Papernot *et al.* (2016a) argue that these security properties are important in the context of the security objectives of ML systems. Barreno *et al.* (2010) identify integrity and availability as security objectives for intrusion detection systems based on ML.

system as expected.

The data and functions to be protected include the training data, an ML model, model input data, model output data, and false prediction data.

For example, consider possible attacks on the training data. If the training data include personal data, attackers should be prevented from accessing the data to protect confidentiality. It is also necessary to prevent alteration of the training data and the ML model if such alteration would degrade system integrity. Finally, from the standpoint of system availability, denial-of-service attacks need to be mitigated. In denial-of-service attacks, large volumes of training data are sent to the model generator to degrade the functioning of the generator's server.

B. Entities' Security Levels and Attacker's Capability

Here, an attacker is a third party outside the ML system with the intention to violate the confidentiality, integrity, and availability of the data and the functions managed or performed by each entity.

We assume that the training-data provider and the system user have moderate security levels that are inadequate, so their data and functions are inferred or altered through unauthorized access to their internal IT systems from the Internet or collusion with entity insiders.[7]

The model generator and operator are assumed to have security levels sufficiently high to prevent an attacker from accessing the data and functions they control, i.e., the training data, the ML algorithm, the ML model, and the model input and output data. These entities are assumed to implement security measures such that, even in the event of an attacker accessing their data and/or functions, the effect would be negligible. For instance, as a countermeasure against the theft of training data, the model generator uses an ML system utilizing homomorphic encryption, as proposed by Dowlin *et al.* (2016) and Phong *et al.* (2018). Such encryption enables the training process to be conducted using encrypted training data. Alternatively, the model generator is able to modify the training data to prevent the identity of individuals and/or organizations from being revealed.[8]

For ML systems in which the organization acting as the model generator or operator also acts as the training-data provider or the system user, the training-data provider or the system user is assumed to implement the advanced security measures adopted by the model generator or operator.

As for the communication channels, we assume that they are securely protected using cryptographic protocols such as Transport Layer Security. This means it would be difficult for an attacker to eavesdrop on or alter data transmitted through the channels. If the attacker obtains cryptographic session keys from the training-data provider or the system user, the attacker is able to eavesdrop on or alter the transmitted data.

.....

7. The training-data provider is assumed to appropriately select the training data pursuant to predetermined processes.
8. Another approach is to utilize privacy-preserving ML based on secret sharing and multi-party computation (Mohassel and Zhang [2017], Mohassel and Rindal [2018]). In this approach, the model generator and operator functions are distributed among multiple organizations. Even if a subset of the organizations is attacked, the training data and the ML model are not disclosed to the attacker.

C. Attacks and Security Measures

Given the security measures mentioned above, we consider that an attacker has the ability to manipulate the data of the training-data provider and/or the system user (i.e., the training data, the model input and output data, and the false prediction data).

1. Manipulation of training data

The attacker attempts to infer training data from the training-data provider and then extract from the data confidential information about individuals and organizations. The attacker also attempts to induce the model generator to generate a malicious model by altering the training data. Such an attack was described by Biggio, Nelson, and Laskov (2011, 2012), Biggio *et al.* (2013), and Goodfellow, McDaniel, and Papernot (2018). The attacker attempts to interrupt the normal operations of the model generator by sending large volumes of training data.

As a countermeasure against the leakage of training data, the training-data provider either uses non-confidential data as training data or modifies the training data in such a way as to make it difficult for the attacker to extract confidential information. As a countermeasure against the generation of a malicious model, the model generator either detects and removes malicious training data before the training process or uses an ML algorithm that mitigates the effect of such data.[9] As a countermeasure against denial-of-service attacks, the model generator makes use of services such as a Contents Delivery Network (CDN) or a network gateway to control access requests from other networks.[10]

2. Manipulation of ML model input and output data

The attacker attempts to infer the ML model and information regarding the training data, as discussed by Ateniese *et al.* (2015), Fredrikson *et al.* (2014), Fredrikson, Jha, and Ristenpart (2015), Shokri *et al.* (2017), and Tramèr *et al.* (2016).[11] The attacker attempts to induce false classification or false inference, as discussed by Kenway (2018), Nguyen, Yosinski, and Clune (2015), Papernot *et al.* (2017b), Sinha, Kar, and Tambe (2016), and Szegedy *et al.* (2014). The attacker also attempts to conduct a denial-of-service attack against the model operator.

The attacker's probability of inferring the ML model or training data will increase if the attacker makes use of confidence values indicating the degree of correctness of the corresponding output data (Fredrikson *et al.* [2014], Fredrikson, Jha, and Ristenpart [2015], and Tramèr *et al.* [2016]). As a countermeasure against such attacks, the model operator sends modified confidence values to the system user instead of the actual ones.[12]

In order to prevent the inference of training data, the model generator uses an ML algorithm such as Private Aggregation of Teacher Ensembles (PATE) (Abadi *et al.* [2017], Goodfellow [2018], and Papernot *et al.* [2017a]). With PATE, the training data are divided into several subsets and used to generate corresponding ML models. A final

.....

9. Several methods have been proposed for detecting and removing malicious input data, including the use of neural networks and principal components analysis (Carlini and Wagner [2017]). A method using input data normalization has been proposed for mitigating the effect of malicious input data.
10. A CDN delivers web contents with low latency by allocating many proxy servers on the Internet. It also controls a high volume of access requests to a certain server.
11. Inference of an ML model has the same effect as stealing the model.
12. For example, the model operator truncates confidence values.

ML model is generated using output data of the models as part of training data.

As countermeasures against the inducement of false predictions by using malicious input data, the model operator detects and removes such data before the prediction process, or the model generator uses a method to mitigate the effect of the malicious input data on the output data. For instance, defensive distillation and adversarial training are useful. In defensive distillation, an initial ML model is generated using given training data and an ML algorithm. The feature values of the training data are input into the initial ML model. A distillated ML model is generated by feeding the same ML algorithm pairs of the feature values and the corresponding output data as new training data.[13] In adversarial training, the training process is performed using adversarial examples that induce false predictions (Szegedy *et al.* [2014]). After these input data are gathered and their labels are corrected, they are fed into the ML algorithm as new training data.

As a countermeasure against denial-of-service attacks, the model operator makes use of services such as a CDN or a network gateway.

3. Manipulation of false prediction data

The attacker attempts to generate false prediction data and sends them to the training-data provider to induce a malicious update of the model. The attacker also attempts to send large volumes of false prediction data to cause a denial-of-service attack.

As a countermeasure against the sending of false prediction data, the training-data provider detects and removes such data before the training process. Alternatively, the model generator uses an ML algorithm that mitigates the effect of malicious training data on the ML model. As a countermeasure against denial-of-service attacks, the training-data provider makes use of services such as a CDN or a network gateway.

D. Relationship between Attacks and Types

Table 2 summarizes the possible adversarial attacks for each type of ML system.

When the organization acting as the model generator or the model operator also acts as both the training-data provider and the system user, we assume that advanced security measures are implemented by all of the entities. In these ML systems (types 8, 11, and 12), since the attacker is not capable of manipulating any data, we do not have to take into account any attacks described in Table 2.

Next, consider situations in which the organization acting as the model generator or the model operator also acts as the training-data provider but not the system user (types 3, 4, and 10). In this case, the attacker is capable of manipulating data and functions controlled by the system user but not those controlled by the training-data provider.

When the organization acting as the model generator or the model operator also acts as the system user but not the training-data provider (types 5 and 9), the attacker makes use of data and functions controlled by the training-data provider but not those controlled by the system user.

When the organization acting as the model generator or the model operator acts as

.....

13. Distillation is a compression method used to reduce the complexity of an ML model. It reduces the number of layers in a neural network and achieves the same level of accuracy. The robustness of the model is thereby strengthened; i.e., the output data do not significantly change even if certain changes are made to the corresponding input data. In defensive distillation (Papernot *et al.* [2016b]), a new model is distillated without reducing the number of layers in order to further improve its robustness.

Table 2 Attacks and Security Measures Assumed in Each Type of ML System

Data available to an attacker	Attacks	Security measures	Corresponding types
Training data	Infer training data	Use non-confidential data as training data	1, 2, 5, 6, 7, 9
	Generate malicious model	Detect and remove malicious training data Use ML algorithm that mitigates effect of malicious training data	
	Conduct denial-of-service attack against model generator	Use CDN and/or network gateway	
Model input and output data	Infer model	Modify confidence values for system user	1, 2, 3, 4, 6, 7, 10
	Infer training data and/or related information	Modify confidence values for system user Use ML algorithm such as PATE	3, 4, 10 (1, 2, 6, 7)
	Induce false classification or inference	Detect and remove malicious input data Apply methods that mitigate effect of malicious input data	1, 2, 3, 4, 6, 7, 10
	Conduct denial-of-service against model operator	Use CDN and/or network gateway	
False prediction data	Generate malicious model	Detect and remove malicious training data Apply methods that mitigate effect of malicious training data	3, 4, 10 (1, 2, 6, 7)
	Conduct denial-of-service against training-data provider	Use CDN and/or network gateway	

Note: For types in parentheses, the attacker places higher priority on attacks using training data.

neither the training-data provider nor the system user (types 1, 2, 6, and 7), the attacker makes use of data and functions controlled by both the training-data provider and the service user. Note that the attacker does not have to make use of data and functions controlled by the system user for these types because the attacker is capable of directly manipulating the training data. Such attacks are more powerful than those against data and functions controlled by the system user.

IV. Security Analysis of Machine Learning Systems Used in Financial Services

A. Overview of Security Measures of ML Systems

When considering security measures for an ML system, the financial institution is able to identify the typical attacks for that type of system by referring to Table 2. Typical attacks are categorized as i) inference of training data and related information, ii) inference of the ML model, iii) generation of a malicious model, iv) inducement of

false classification or false inference, and v) denial-of-service. Since several security measures have already been established for denial-of-service attacks, the main task for financial institutions using ML systems is to identify which attacks among i) to iv) above should be considered and how to address them. When determining which approach to take, financial institutions should estimate the effect (or economic loss) of a successful attack. If the estimated effect or loss is acceptable, no additional measures are required. Otherwise, security measures need to be implemented.

Regarding inference of training data and related information, it is necessary to assess data leakage risks. If the training data include only public data such as financial market data and statistical data, the risk is low because the training data are not confidential. On the other hand, if the training data include confidential information such as customers' assets and financial transactions, the risk is assessed as high. In such a case, the financial institutions should implement additional security measures to protect the training data.

Regarding inference of the ML model, financial institutions should focus on the risk of model leakage. Suppose that an ML system is used to forecast financial markets. If the institution regards the system as a valuable asset, it should implement security measures to protect the model. Alternatively, if an ML system is used to automatically reply to customer inquiries for which the information is not confidential, the effect of model leakage is regarded as negligible. In this case, the institution does not need to implement additional security measures.

If the training data include confidential information, the financial institution should implement appropriate security measures to prevent the attacker from inferring the training data and related information.

Regarding the generation of a malicious model, financial institutions should consider the effect (economic loss) of a successful attack. For a system sending automated replies to customer inquiries, the loss caused by a malicious model is regarded as negligible. This is because bank employees follow up on and correct inappropriate responses. On the other hand, for an ML system forecasting financial market developments, false prediction leads to inappropriate asset management. Financial institutions need to implement security measures appropriate for the risk. The same consideration applies to attacks inducing false classification or false inference.

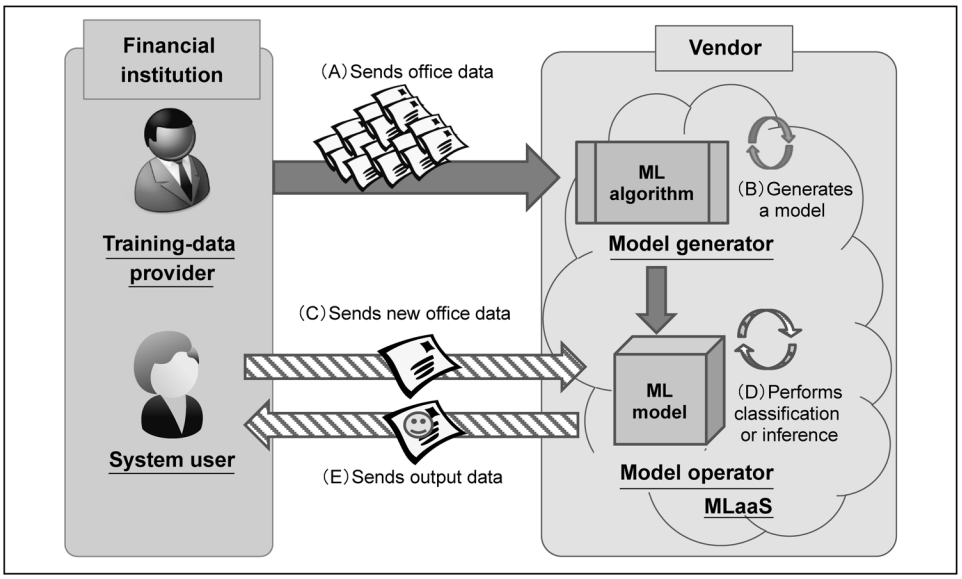
B. Typical Use Cases of ML Systems in the Financial Sector

The use of ML systems is being broadly explored in the financial sector as a means to i) enhance the efficiency of business operations, ii) improve the quality of financial services, iii) assist decision making and prediction, and iv) manage and reduce risks. In this subsection, we present typical use cases of ML systems for each of these purposes. We then discuss attacks and security measures for each use case by referring to Table 2.

1. Enhance efficiency of business operations

Many of the business operations of financial institutions require special expertise, as well as consistency with past operations. The expertise needed for such operations as payment of insurance claims, drafting of loan contracts, and use of optical character readers for bank transfers is not easy to share among employees because the training required is time-consuming and costly. It is also difficult to conduct these operations

Figure 2 Overview of ML System Developed Using MLaaS



with existing IT systems. Since ML is capable of finding unknown similarities and rules from existing data, ML systems partially perform these tasks.

MLaaS is used to develop such ML systems. For example, an ML system could be developed as follows (see Figure 2):

(A) The financial institution (as the training-data provider) sends the vendor (as the model generator) office data for use as training data.
(B) The vendor generates an ML model by using the office data.
(C) The financial institution (as the system user) sends the vendor (as the model operator) new office data to be input into the ML model.
(D) The vendor performs classification or inference by inputting the office data into the ML model.
(E) The vendor sends the financial institution output data generated by the ML model.

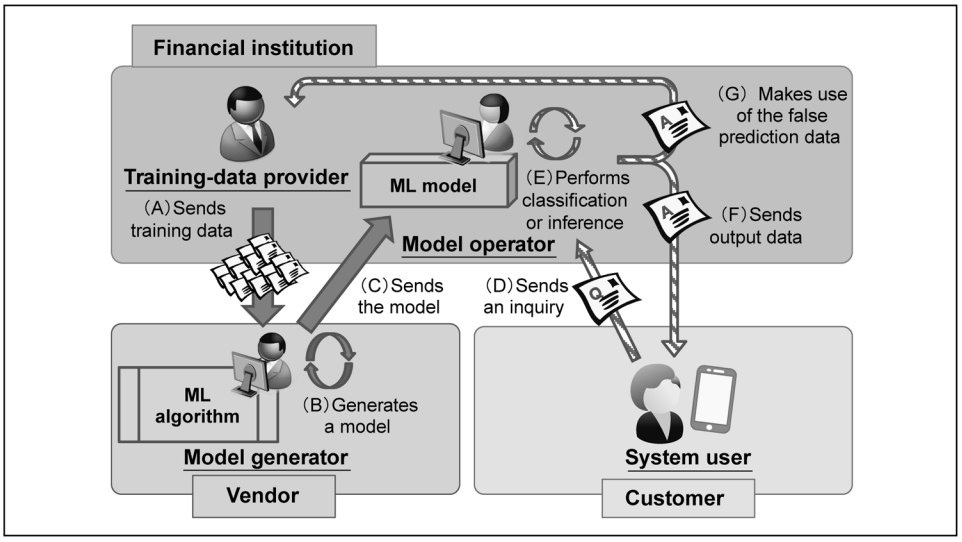
The financial institution acts as both the training-data provider and the system user while the vendor acts as both the model generator and the model operator. This example corresponds to a type-7 ML system.

With a type-7 ML system, an attacker accesses data and functions controlled by the training-data provider and/or the system user. The financial institution should prevent leakage of training data if confidential information such as customers' personal data is included. To reduce the risk of leakage, the use of data modification methods to remove confidential information is recommended.[14]

The financial institution should also be alert to attacks that carry out inference of the model, generation of a malicious model, or inducement of false predictions. While

14. Modification of training data lowers the performance of the ML model. Therefore, it is necessary to consider how to maintain the performance level when selecting the data modification method.

Figure 3 Overview of ML System for Chatbot



the effect of model inference is negligible for systems deployed to improve business operations, the generation of a malicious model and inducement of false predictions degrade the reliability of such operations. As a countermeasure as such attacks, financial institutions should implement a method to detect and remove malicious training and input data as well as appoint someone to check the output data.

2. Improve quality of financial services

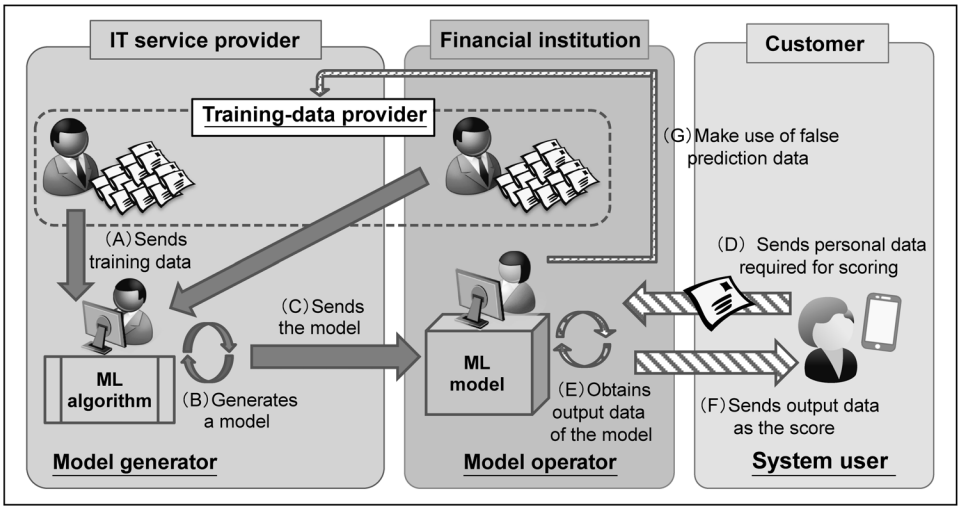
Several financial institutions have already introduced chatbot-based services to improve the quality of customer-related operations.[15] A chatbot is used as a tool to communicate with customers on various platforms such as smartphone-based applications and web browsers. A chatbot automatically sends consistent and reliable information to customers in response to their inquiries by taking into account the customers' attributes. It is also used to propose financial products.

A basic function of a chatbot is to reply to a predetermined set of inquiries. Various ML algorithms for chatbots are being provided by IT vendors. These algorithms are generally based on the following processes (see Figure 3).

(A) The financial institution (as the training-data provider) sends the vendor (as the model generator) training data generated from office data including query history, product information, and customer characteristics.
(B) The vendor generates a model by inputting the training data into an ML algorithm for a chatbot.
(C) The vendor sends the financial institution (as the model operator) the model.

15. New smartphone-based services have been launched such as those by Bank of America Corporation (https://promo.bankofamerica.com/erica/) and Capital One Bank (USA), N.A. (https://www.capitalone.com/applications/eno/). These services provide support for customer financial activities, malicious transaction warnings based on an analysis of the customer's transaction records, as well as account balance information. In addition, a service running as a chatbot on social network service (SNS) has been launched that provides information about insurance products and insurance premiums (https://www.lifenet-seimei.co.jp/line).

Figure 4 Overview of ML System for Credit Scoring of Personal Loans



(D) A customer (as a system user) sends the financial institution an inquiry through a smartphone application or SNS.
(E) The financial institution inputs the inquiry as data into the model, which then performs classification or inference.
(F) The financial institution sends the customer the output data.
(G) The financial institution makes use of the output data as false prediction data if the output data are incorrect.

The financial institution acts as the training-data provider and the model operator. The vendor and the customer act as the model generator and the system user, respectively. This corresponds to a type-4 ML system.

With a type-4 ML system, an attacker attempts to manipulate data and functions controlled by the system user. If the service is limited to replying to customer inquiries on the basis of public information or to inform customers of financial products, the training data do not include confidential information. Even if the attacker successfully inferred the training data and/or the model, doing so would not lead to a serious incident, such as the leakage of personal data.

Table 2 also shows that an attacker induces false predictions by sending malicious input data to the model or generates a malicious model by manipulating false prediction data. If doing so results in the frequent occurrence of inappropriate responses, the financial institution's reputation is harmed. Hence, financial institutions should regularly check any false prediction data.

3. Assist decision making and prediction

Credit scoring for screening of loan applicants is another main application of ML systems.[16] To develop a high-quality ML system for credit scoring of personal loans, it is

16. Credit scoring services for personal loans are already being provided through web applications. The training data include data on various customers' attributes such as their characters, hobbies, and behavior patterns during web shopping, as well as their ages, incomes, and affiliations. These data are also input to the model during the prediction process. A credit score is generated on the basis of such information.

crucial to collect a large volume of customers' personal data. For instance, financial institutions collaborate with IT service providers, which hold a broad range of customer data and have expertise in developing ML systems. For this case, the following processes are assumed (see Figure 4).

(A) The financial institution (as the training-data provider) sends the IT service provider (as the model generator) customers' personal data.
(B) The IT service provider (as the training-data provider) creates training data by using the data sent by the financial institution as well as its own customers' data. The IT service provider (as the model generator) then generates the ML model.
(C) The IT service provider sends the financial institution (as the model operator) the ML model.
(D) The customer (as the system user) sends the financial institution her/his personal data required for the scoring through a smartphone application or SNS.
(E) The financial institution inputs the data into the model and obtains the corresponding output data.
(F) The financial institution sends the customer the output data as the score.
(G) The financial institution and the IT service provider make use of the output data as false prediction data if they are incorrect.

While the financial institution acts as the model operator, the IT service provider acts as the model generator. Both of them also act as the training-data provider. The customer acts as the system user. This corresponds to a type-3 ML system.[17]

With a type-3 ML system, an attacker manipulates data and functions controlled by the system user. By obtaining the model input and/or output data, the attacker infers the training data and the model. The attacker also induces an incorrect score by sending the financial institution malicious input data.

It is difficult to limit the attacker's access to the input and output data because financial institutions generally permit any individual to use the service. Therefore, financial institutions should consider security measures to prevent the leakage of confidential information under the assumption that an attacker succeeds in obtaining the input and output data. For example, if the training data include numeric personal data such as customer ages or annual incomes, they should be modified by using truncation or rounding methods to make their inference more difficult.[18]

If the attacker is successful in inducing incorrect credit scores, the screening of loan applications will not be appropriately conducted, and the institution's credit risk will increase. Moreover, if manipulated credit scores are fed into the model, the model will be incorrectly updated and then generate incorrect credit scores for other customers. As a countermeasure against such attacks, financial institutions should implement techniques to detect and remove malicious training and input data and/or adopt an ML

17. Alternatively, the financial institution and the IT service provider may establish a joint venture to launch a credit scoring service. In that case, the joint venture acts as the training-data provider, the model generator and the model operator. This corresponds to a type-10 ML system, as exemplified by J.Score CO., LTD. (2017).
18. For example, customer ages are rounded to the nearest ten and customer incomes are rounded to the nearest million (for yen) for use as training data.

algorithm that mitigates the effect of such attacks.

4. Manage and reduce risks

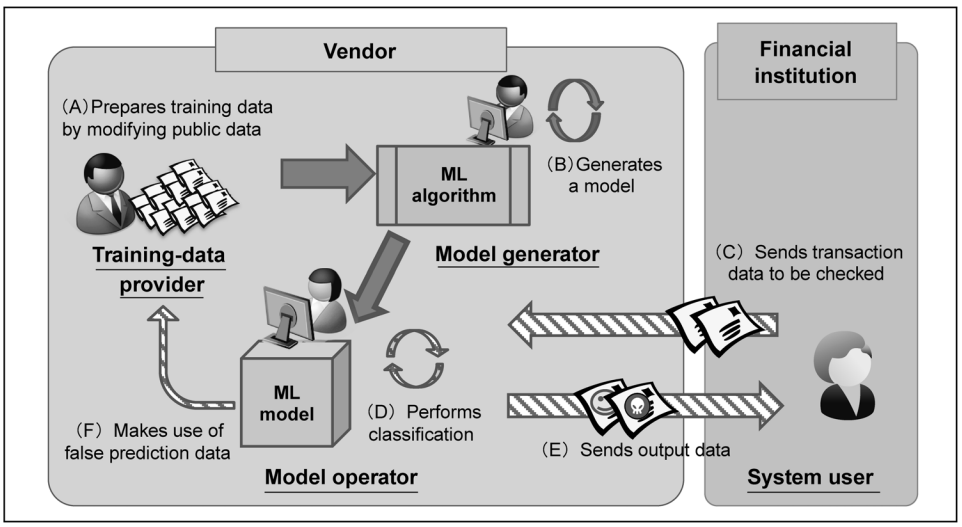
Financial institutions are also developing ML-based anomaly detection systems as tools to reduce operational risks. Typical applications include detection of anomalies in financial markets and detection of fraudulent credit card transactions. For the former, market data such as transaction orders, market liquidity, and price fluctuations are generally used as training data. For the latter, data on past fraudulent transactions are generally used as training data.[19]

Suppose that an IT vendor provides a financial institution with an anomaly detection service based on a model generated using public data on financial markets as training data. For this case, the following processes are assumed (see Figure 5).[20]

(A) The vendor (as the training-data provider) prepares training data by modifying public data.
(B) The vendor (as the model generator) uses the training data to generate an ML model.
(C) The financial institution (as the system user) sends the vendor (as the model operator) transaction data to be checked.
(D) The vendor inputs the transaction data into the model and performs the classification.
(E) The vendor sends the output data to the financial institution.
(F) The vendor makes use of the model input and output data as false prediction data if they are incorrect.

The vendor acts as the training-data provider, the model generator, and the model

Figure 5 Overview of ML System for Anomaly Detection



19. Cloud-based services for detecting fraudulent transactions are already being provided for financial institutions.
20. There could also be cases in which financial transaction data held by a financial institution are used as training data. Since the financial institution acts as the training-data provider, this corresponds to a type-7 ML system.

operator while the financial institution acts as the system user. This corresponds to a type-10 ML system.

With a type-10 ML system, an attacker makes use of data and functions controlled by the financial institution by impersonating it to the vendor or by colluding with someone within the financial institution. The attacker infers the training data and/or model by using the model input and output data. The attacker also induces incorrect output data and/or the generation of a malicious ML model by feeding malicious input data into the model.

Since public data are used as the training data, inference of the training data is not a serious problem. Inference of the model is also not a concern to the financial institution.

The effects of the other attacks depend on the model's purpose. For models used to detect anomalies in financial markets, incorrect detection leads to inappropriate execution of financial transactions and thus could cause non-negligible financial losses. For models used to detect fraudulent transactions, an attacker modifies the input data so that detection is prevented. If the model were updated using the resulting incorrect output data as false prediction data, the updated model would generate manipulated output data. Financial institutions should implement methods to detect and remove malicious input data and false prediction data and/or adopt an ML algorithm that mitigates the effect of such attacks.

V. Concluding Remarks

The use of ML systems in the financial sector is still at an early stage. As far as the authors are aware, critical security incidents regarding such systems have not yet been reported. Nevertheless, it is important to understand the vulnerabilities and security risks specific to ML systems and to discuss how to deal with them in advance.

In this paper, we classified ML systems into twelve types on the basis of the relationships among entities involved in the system. When deploying an ML system, it is useful to identify its type and analyze its vulnerabilities. The financial institution should then assess the potential effects of these vulnerabilities being exploited, taking into account the criticality of the data managed by each entity, and implement appropriate protection methods.

Information technologies are steadily advancing while, at the same time, adversarial techniques are becoming more sophisticated. In order to implement appropriate security measures, financial institutions need to be cognizant of new developments in both adversarial and defensive techniques and update their security measures as necessary.

References

Abadi, Martin, Úlfar Erlingsson, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Nicolas Papernot, Kunal Talwar, and Li Zhang, "On the Protection of Private Information in Machine Learning Systems: Two Recent Approaches," *Proceedings of IEEE Computer Security Foundations Symposium 2017*, IEEE, 2017, pp. 1–6.

Ateniese, Giuseppe, Luigi V. Mancini, Angelo Spognardi, Autonio Villani, Domenico Vitali, and Giovanni Felici, "Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers," *International Journal of Security and Networks*, 10(3), Inderscience Publishers, 2015, pp. 137–150.

Barreno, Marco, Blaine Nelson, Anthony D. Joseph, and J. Doug Tygar, "The Security of Machine Learning," *Machine Learning*, 81(2), Springer-Verlag, 2010, pp. 121–148.

Biggio, Battista, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli, "Evasion Attacks against Machine Learning at Test Time," *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases 2013 Part 3, Lecture Notes in Computer Science*, 8190, Springer-Verlag, 2013, pp. 387–402.

———, Blaine Nelson, and Pavel Laskov, "Support Vector Machines under Adversarial Label Noise," *Proceedings of Asian Conference on Machine Learning*, *Proceeding of Machine Learning Research*, 20, Microtome Publishing, 2011, pp. 97–112.

———, ———, and ———, "Poisoning Attacks against Support Vector Machines," *Proceedings of International Conference on Machine Learning (ICML) 2012*, Omnipress, 2012, pp. 1467–1474.

Carlini, Nicholas, and David Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," *Proceedings of ACM Workshop on Artificial Intelligence and Security (AISec) 2017*, Association for Computing Machinery, 2017, pp. 3–14.

Dowlin, Nathan, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing, "CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy," *Proceedings of International Conference on Machine Learning (ICML) 2016*, *Proceedings of Machine Learning Research*, 48, Microtome Publishing, 2016, pp. 201–210.

Fredrikson, Matthew, Somesh Jha, and Thomas Ristenpart, "Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures," *Proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS) 2015*, Association for Computing Machinery, 2015, pp. 1322–1333.

———, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart, "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing," *Proceedings of USENIX Security Symposium 2014*, Advanced Computing Systems Association, 2014, pp. 17–32.

Goodfellow, Ian, "Security and Privacy of Machine Learning," presentation at RSA Conference 2018, RSA, 2018 (available at https://www.iangoodfellow.com/slides/2018-04-rsa.pdf).

———, Patrick McDaniel, and Nicolas Papernot, "Making Machine Learning Robust against Adversarial Inputs," *Communications of the ACM*, 61(7), Association for Computing Machinery, 2018, pp. 56–66.

J.Score CO., LTD., "Mizuho Ginko to Softbank no Gobengaisha J. Score ga Nihonhatsu no Fintech Service, AI Score Lending, wo Honjitsu yori Teikyokaishi (J. Score, a Joint Venture of Mizuho Bank and Softbank, Has Just Started AI Score Lending as the First Fintech Service)," J.Score CO., LTD., 2017 (available at https://www.jscore.co.jp/company/news/2017/0925_01/, in Japanese).

Kenway, Richard, "Vulnerability of Deep Learning," arXiv: 1803.06111v1, 2018.

Mohassel, Payman, and Peter Rindal, "ABY^3: A Mixed Protocol Framework for Machine Learning," *Proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS) 2018*, Association for Computing Machinery, 2018, pp. 35–52.

———, and Yupeng Zhang, "SecureML: A System for Scalable Privacy-Preserving Machine Learn-

ing," *Proceedings of IEEE Symposium on Security and Privacy (SP) 2017*, IEEE, 2017, pp. 19–38.

Nguyen, Anh, Jason Yosinski, and Jeff Clune, "Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*, IEEE, 2015, pp. 427–436.

NTT DATA Corporation, "AI wo Katsuyoshita Chatbot no Shikoteikyo wo Kaishi (NTT Data Has Started Trial Offer of AI-Based Chatbot)," NTT DATA Corporation, 2017 (available at http://www.nttdata.com/jp/ja/news/services_info/2017/2017060901.html, in Japanese).

Papernot, Nicolas, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar, "Semi-Supervised Knowledge Transfer for Deep Learning from Private Training Data," presentation at the International Conference on Learning Representations (ICLR) 2017, OpenReview.net, 2017a (available at https://openreview.net/pdf?id=HkwoSDPgg).

———, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami, "Practical Black-Box Attacks against Machine Learning," *Proceedings of ACM on Asia Conference on Computer and Communications Security (ASIACCS) 2017*, Association for Computing Machinery, 2017b, pp. 506–519.

———, ———, Arunesh Sinha, and Michael Wellman, "Towards the Science of Security and Privacy in Machine Learning," arXiv: 1611.03814v1, 2016a.

———, ———, Xi Wu, Somesh Jha, and Ananthram Swami, "Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks," *Proceedings of IEEE Symposium on Security and Privacy (SP) 2016*, IEEE, 2016b, pp. 582–597.

Phong, Le Trieu, "Privacy-Preserving Stochastic Gradient Descent with Multiple Distributed Trainers," *Proceedings of International Conference on Network and System Security (NSS) 2017*, *Lecture Notes in Computer Science*, 10394, Springer-Verlag, 2017, pp. 510–518.

———, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai, "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption," *IEEE Transactions on Information Forensics and Security*, 13(5), IEEE, 2018, pp. 1333–1345.

Shokri, Reza, Marco Stronati, Congzheng Song, and Vitaly Shmatikov, "Membership Inference Attacks against Machine Learning Models," *Proceedings of IEEE Symposium on Security and Privacy (SP) 2017*, IEEE, 2017, pp. 3–18.

Sinha, Arunesh, Debarun Kar, and Milind Tambe, "Learning Adversary Behavior in Security Games: A PAC Model Perspective," *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS) 2016*, International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 214–222.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing Properties of Neural Networks," Proceedings of International Conference on Learning Representations (ICLR) 2014, arXiv: 1312.6199v4, 2014.

Tramèr, Florian, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart, "Stealing Machine Learning Models via Prediction APIs," *Proceedings of USENIX Security Symposium 2016*, Advanced Computing Systems Association, 2016, pp. 601–618.

Une, Masashi, "Kikaigakushu System no Security ni Kansuru Kenkyudoko to Kadai (Research Trends and Topics on Security of Machine Learning Systems)," *Kin'yu Kenkyu* (Monetary and Economic Studies), 38(1), Institute for Monetary and Economic Studies, Bank of Japan, 2019, pp. 97–124 (in Japanese).

Yoshioka, Nobukazu, "Kikaigakushu System ga Security ni Deautoki (When Machine Learning Systems Meet Security)," *Proceedings of Machine Learning Systems Engineering (MLSE) Workshop 2018*, Special Interest Group on Machine Learning Systems Engineering, 2018, pp. 49–53 (in Japanese).