Elevation of MLsec

by Kantega

Contents

- **4** instruction cards
- **1** card about licenses and references
- **1** contents card
- **52** playing cards in **4** ML threat card “suits”:
  a. Dataset risk: 2-10, J, Q, K, Ace
  b. Model risk: 2-10, J, Q, K, Ace
  c. Input risks 2-10, J, Q, K, Ace
  d. Output risk: 2-10, J, Q, K, Ace
- **2** appendix cards with the BIML Machine learing risk framework
- **8** summary cards

Instructions (1)
for Elevation of MLsec

Elevation of MLsec is an unofficial Machine Learning Security (MLsec) extension of the Elevation of Privilege threat modeling card game. These playing cards portray risks associated with Machine Learning systems that have been identified by research groups. You may play this with or without the original Elevation of Privilege deck. See Instructions (2-3) on how to play.

This work is mainly based on Berryville Institute for Machine Learnings (BIML)'s architectural risk analysis for machine learning systems (BIML-78): berryvilleiml.com and their LLM analysis (BIML-LLM24). We have also added a few relevant LLM specific threats from OWASP's TOP 10 list for Large Language Models found on owasp.org. Risks originating from BIML's ML risk analysis for ML are tagged with [<component label>:<risk number>:<descriptor>]. Risks from OWASP are tagged with [OWASP-[project]-<number>], so for LLMs it's [OWASP-LLM-<number>]. Risks originating from the BIML Architectural Risk Analysis of LLMs are tagged with [LLM:<component label>:<risk number>:<descriptor>].

Instructions (2)
for Elevation of MLsec

Rules
Draw a diagram of the system that you threat model before dealing cards.
Play like “Spades”, where everyone must play in the same suit or call pass, and the highest “bidder” takes the trick and may start the next round. In Elevation of MLsec, cards in the **Dataset risk** suit are trump. Use a diagram / model of an ML system, ML lifecycle or a system with an ML component. The player should be able to explain how the risk relates to the model. In case of an argument, when something is not clear from the diagram, the player may make the necessary assumptions about the system to call a risk.

The ace of each suit is an open threat card. When played, the player must identify a threat not listed on another card. To help with this, see the summary cards in the back of the deck.

Points: 1 for a threat on your card, +1 for taking the trick.

When all the cards have been played, whoever has the most points wins. Don’t forget to triage the threats you discovered, and lastly remember to have fun.

Kantega

Instructions (3)
for Elevation of MLsec

Setup

1. Draw a system model where everyone can see it.
2. Ask everyone to set aside skepticism for 15 minutes.
3. Setup details
    a. Goal: Win hands by connecting the threat on the card to the system and playing the highest card.
    b. Play in small groups (3-6 people), using one deck per group. Remove the front cards (instructions, strategy) and the back (lists of threats), so you only have threat cards, and shuffle those. Deal the whole deck to the players.
    c. Determine the first player based on who has the lowest card of Input risks. Players following have to play the same suit.
4. Play!
    a. Make sure people take notes on the threats
    b. Some people will play collaboratively with their hands face up in front of them, others competitively, with cards close to their chest. Elevation of Privilege, play it

Instructions (4)
for Elevation of MLsec

We divide our threat categories into four categories, which represent the four card suits in the deck. The ten components from the BIML-78 risk analysis are mapped to our four suits. Generally, the mapping is as follows: **Dataset risks** (1. Raw, 2. Training, 3. Assembly), **Model risks** (4. Algorithm, 5. Evaluation, 7. Model), **Input risks** (6. Inputs) and **Output risks** (9. Outputs). The inference risks (8. Inference) and the system-wide risks (10. System) are applied to categories where the individual risk fits best. Where appropriate, BIML risks from other components than the mapping described above have been mapped to our four categories. Also, a few LLM risks have been added to give the deck a little flavor, even if they mostly are actually covered by the generic model. See Appendix 1 for more details about the BIML Architectural Risk Model of Machine Learning systems, or download the whole model and paper from their website.

Working group: Elias Brattli Sørensen, Jorun Kristin Bremseth, Emil Jakobus Schroeder, Edvard Kristoffer Karlsen.

Kantega

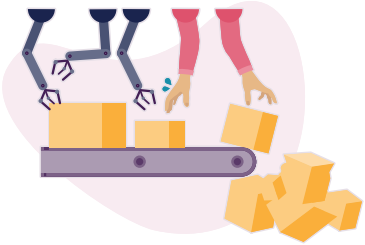
Model risk

2

Catastrophic forgetting

[eval:5:catastrophic forgetting]

When a model is filled with too much overlapping information, collisions in the representation space may lead to the model “forgetting” information.



Kantega

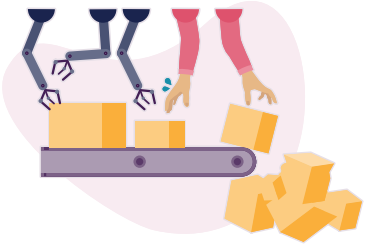
Model risk

3

Oscillation

[alg:8:oscillation]

An ML system may end up oscillating and not properly converging if using gradient descent in a space with a misleading gradient.



Kantega

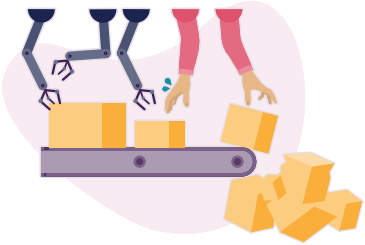
Model risk

4

Randomness

[alg:4:randomness]

Setting weights and thresholds with a bad RNG can damage system behavior and lead to subtle security issues.



Kantega

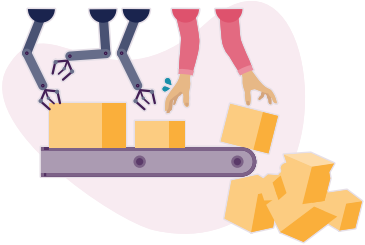
Model risk

5

Online system manipulation

[alg:1:online]

When an ML system system online keeps learning during operations, clever attackers can nudge the model so that it drifts from its intended operational profile.



Kantega

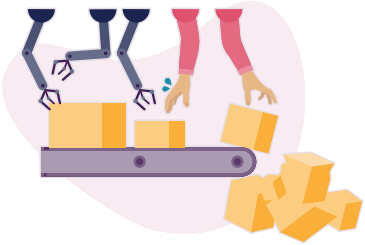
Model risk

6

Overfitting

[eval:1:overfitting]

The model learns its training dataset so well that it's no longer able to generalize outside of the training set and will perform poorly.



Kantega

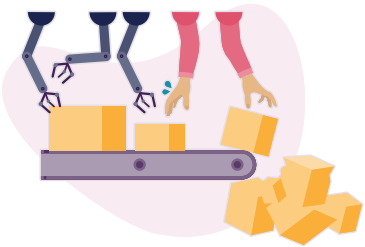
Model risk

7

Hyperparameters

[inference:3:hyperparameters]

An attacker that can control the hyperparameters can manipulate the future training of the machine learning model



Kantega

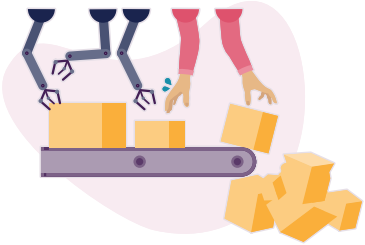
Model risk

8

Hosting

[inference:4:hosting]

The server where the model is hosted is insufficiently protected against unauthorized parties.



Kantega

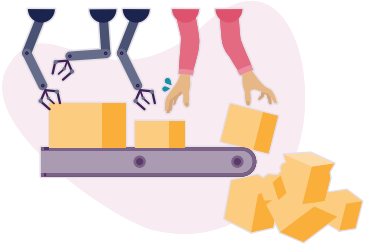
Model risk

9

Hyperparameter sensitivity

[alg:10:hyperparameter sensitivity]

Sensitive hyperparameters that have been set experimentally may not be sufficient for the intended problem space, and can lead to overfitting.



Kantega

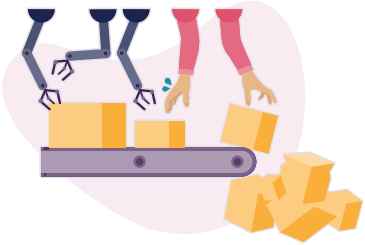
Model risk

10

Model theft

[model:5:steal the box]

Stealing ML system knowledge is possible through direct input/output observation, enabling attackers to reverse engineer the model.



Kantega

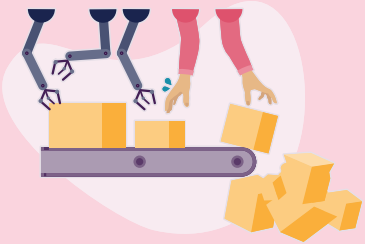
Model risk

J

Training set reveal

[model:4:training set reveal]

Most ML algorithms learn a great deal about its data and store a representation internally. This data may be sensitive, and can potentially be extracted from the model.



Kantega

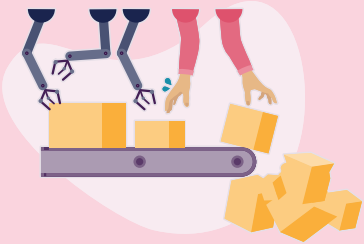
Model risk

Q

Trojanized model

[model:2:Trojan]

Model transfer leads to the possibility that what is being reused may be a Trojaned (or otherwise damaged) version of the model.



Kantega

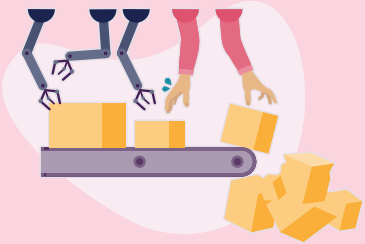
Model risk

K

Improper re-use of model

[model:1:improper re-use]

ML models are re-used in transfer situations, where a pre-trained model is specialized toward a new use case. The model may be transferred into a problem space it's not designed for.

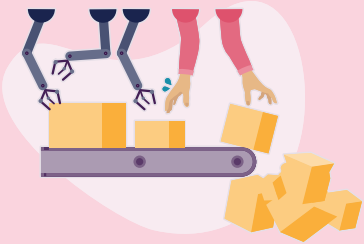


Kantega

Model risk

A

You have invented your own risk associated with machine learning models.



Kantega

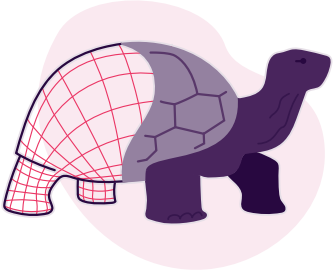
Input risk

2

LLM feedback scores

[LLM:inference:6:feedback scores]

Some LLM chat systems allow user feedback as a parameter for tuning their system. This can be abused by attackers that give feedback in a coordinated fashion to nudge the ML system.



Kantega

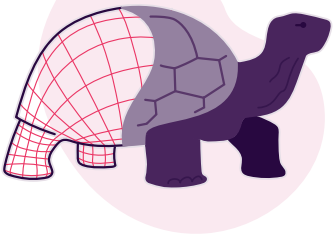
Input risk

3

Open to the public

[LLM:input:3:open to the public]

An LLM model is often open to the public, which makes it susceptible to attacks from users.



Kantega

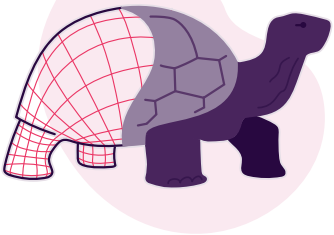
Input risk

4

Sponge input

[LLM:input:5:sponge input]

A sponge attack provides an LLM system with input that is more costly to process than “normal”. Like a DoS attack, as it seeks to exhaust processing budget.



Kantega

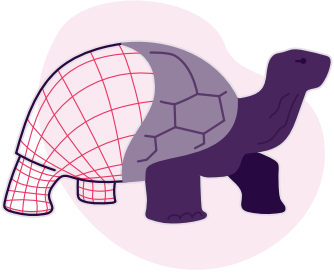
Input risk

5

Input ambiguity

[LLM:input:6:input ambiguity]

English, the main interface language for LLMs, is an ambiguous interface. Natural language can be misleading, making LLMs susceptible to misinformation.



Kantega

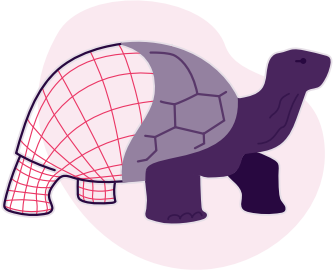
Input risk

6

Text encoding

[raw:7:text encoding]

An ML system engineered with one text encoding scheme in mind might yield surprising results if presented with a differently encoded text.



Kantega

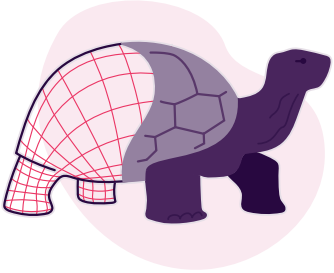
Input risk

7

Denial of service

[system:10:denial of service]

Denial of Service attacks can have a massive impact on a critical ML system. When an ML system breaks down, recovery may not be possible.



Kantega

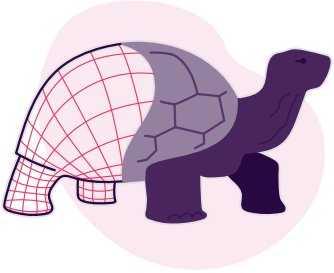
Input risk

8

User risk

[inference:5:user risk]

A user may expose their personal data and their interests to the owners of an ML system when they interact with the system.



Kantega

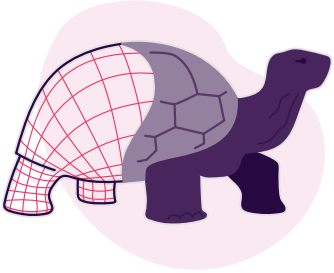
Input risk

9

Dirty input

[input:3:dirty input]:

Dirty inputs can be hard to process, and may be leveraged by an attacker adding noise in their prompts or in data sources for future training.



Kantega

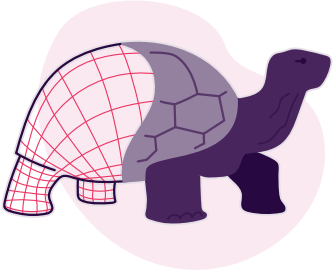
Input risk

10

Controlled input stream

[input:2:controlled input stream]

Outside sources of input may be manipulated by an attacker.



Kantega

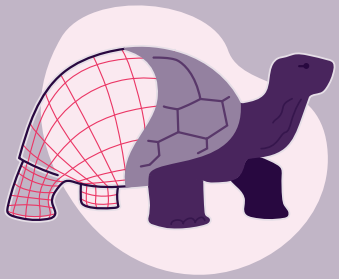
Input risk

J

Looped input

[input:4:looped input]

ML system output to the real world may feed back into training data or input, leading to a feedback loop, termed recursive pollution.



Kantega

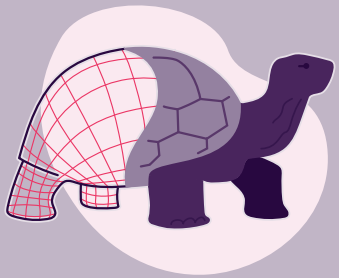
Input risk

Q

Prompt injection

[LLM:input:2:prompt injection]

Input manipulation for LLMs. An attacker manipulates a large language model (LLM) through malicious inputs to override initial instructions given in system prompts.



Kantega

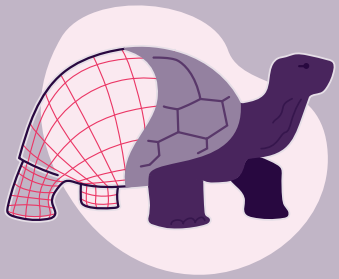
Input risk

K

Malicious input

[input:1:adversarial examples]

Fool a machine learning system by providing malicious input that causes the ML system to make a false prediction or categorization.

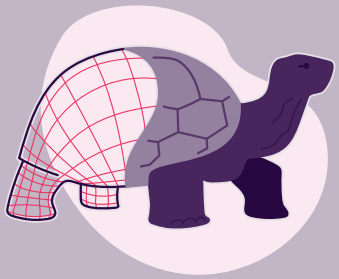


Kantega

Input risk

A

You have invented your own risk associated with machine learning input.



Kantega

Output risk

2

Cry wolf

[system:6:cry wolf]

If an ML model is integrated into a security decision and raises too many alarms, its output may be ignored.

Kantega

Output risk

3

Black box discrimination

[system:1:black box discrimination]

ML systems that operate with high impact decisions based on personal data carry the risk of illegal discrimination based on bias.

Kantega

Output risk

4

LLM overreliance

[OWASP LLM09]

Dependence on an LLM without oversight may lead to misinformation and legal concerns. It will also be hard to detect an attack against the LLM system.

Kantega

Output risk

5

Inscrutability

[output:4:inscrutability]

In far too many cases with ML, nobody is really sure how the trained systems do what they do. This negatively affects trustworthiness.

Kantega

Output risk

6

Miscategorization

[output:3:miscategorization]

Bad output due to internal bias, malicious input or other attacks may escape into the world.

Kantega

Output risk

7

Transparency

[output:5:transparency]

It is easier to perform attacks undetected on a black-box system which is not transparent about how it works.

Kantega

Output risk

8

Confidence scores

[inference:3:confidence scores]

An ML model's confidence scores can help an attacker tweak inputs to make the system misbehave.

Kantega

Output risk

9

Wrongness

[LLM:output:2:wrongness]

LLMs are stochastic in their nature, and can generate highly convincing misinformation in their attempt to satisfy the prediction of the next tokens from a prompt.

Kantega

Output risk 10

Excessive LLM agency

[OWASP LLM08]

An LLM-based system may undertake actions leading to unintended consequences if granted excessive functionality, permissions, or autonomy.

Kantega

Output risk

J

Overconfidence

[system:2:overconfidence]

An ML model integrated into a system with its output treated as high confidence data may cause a range of unexpected issues.

Kantega

Output risk

Q

Error propagation

[system:5:error propagation]

When ML output is input to a larger decision process, errors in the ML subsystem may propagate in unforeseen ways.

Kantega

Output risk

K

Output manipulation

[output:1:direct]

An attacker directly manipulates the output stream getting between the ML system and its receiver. This may be hard to detect because models are sometimes opaque.

Kantega

Output risk

A

You have invented your own risk associated with machine learning output.

Kantega

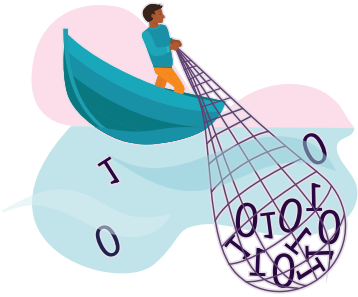
Dataset risk

2

Metadata

[raw:10:metadata]

Metadata may accidentally degrade generalization since a model learns a feature of the metadata instead of the content itself.



Kantega

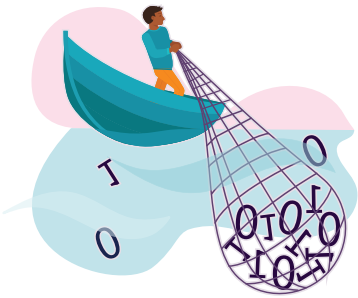
Dataset risk

3

Data rights

[LLM:raw:4:data rights]

Copyrighted, privacy protected or otherwise legally encumbered data are scraped from the internet to train ML models. This can lead to expensive legal entanglements.



Kantega

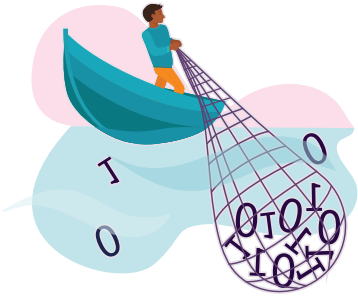
Dataset risk

4

Partitioning

[assembly:4:partitioning]

Bad data partitions for training, validation and testing datasets may lead to a misbehaving ML system.



Kantega

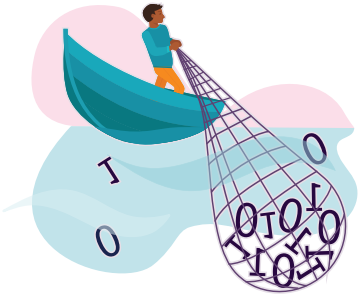
Dataset risk

5

Normalization

[assembly:3:normalize]

Normalization changes the nature of raw data, and may destroy the feature of interest by introducing too much bias.



Kantega

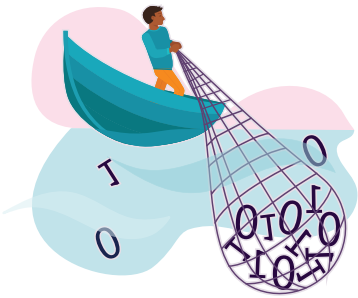
Dataset risk

6

Annotation

[assembly:2:annotation]

The way data is annotated into features can be directly attacked, introducing attacker bias into a system.



Kantega

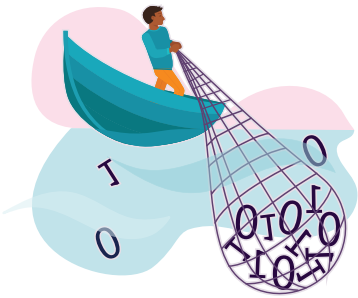
Dataset risk

7

Encoding integrity

[assembly:1:encoding integrity]

Pre-processing and encoding of the data can lead to encoding integrity issues if the data has bias or discrimination in its nature.



Kantega

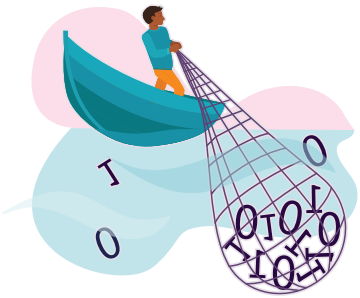
Dataset risk

8

Bad evaluation data

[eval:2:bad eval data]

A bad evaluation dataset can give unrealistic projections to how the model will perform when it is shipped to production.



Kantega

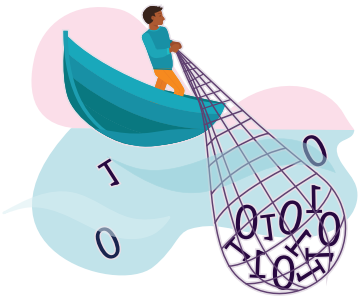
Dataset risk

9

Storage

[data:4:storage]

Data may be stored and managed insecurely. Who has access to the data, and why?



Kantega

Dataset risk 10

Recursive pollution

[LLM:raw:1:recursive pollution]

An ML model (LLM or other) generates incorrect content that content finds its way into future training data, which can damage the accuracy and reliability of the model.

Kantega

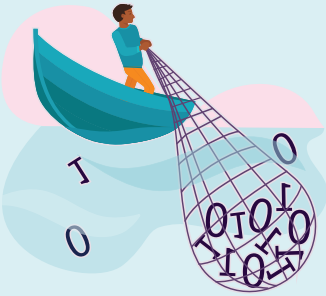
Dataset risk

J

Data integrity

[system:2:data integrity]

If distributed datasets do not have proper integrity checks in place, data can be tampered with undetected as it passes between components.



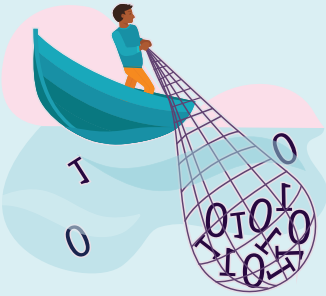
Kantega

Dataset risk

Data confidentiality

[raw:1:data confidentiality]

Sensitive and confidential data that is used for ML training can be disclosed with extraction attacks.



Kantega

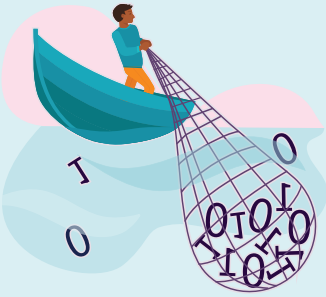
Dataset risk

K

Data poisoning

[data:1:poisoning]

An attacker intentionally manipulates data to disrupt, introduce bias, control or otherwise influence ML training. On the internet, lots of data are already poisoned “by default”.

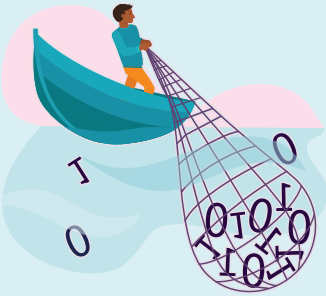


Kantega

Dataset risk

A

You have invented your own risk associated with machine learning datasets.



Kantega

Appendix 1
for Elevation of MLsec

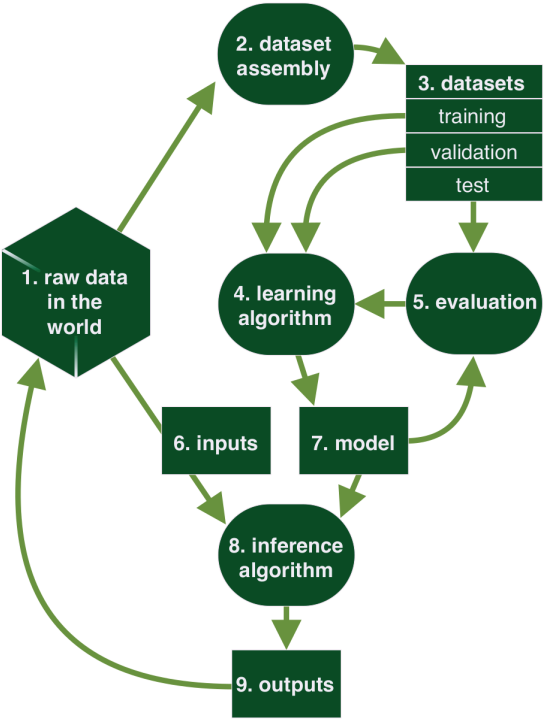


Figure 1: BIML's generic ML lifecycle model

Appendix 1 cont.
for Elevation of MLsec

The nine components of BIML's generic ML lifecycle model (Figure 1) map in a straightforward way into specific ML models. Example, Google's Neural Machine Translation model (GNMT):

1. Raw data in the world. GNMT makes use of numerous Google internal datasets.
2. Dataset assembly. Raw text is organized into sentence pairs between two languages.
3. Datasets. The parsed text pairs are separated into a training set and test set.
4. Learning algorithm. At a high level, GNMT's learning algorithm consists of an Encoder RNN, an attention module, and a Decoder RNN.
5. Evaluation. Training refined with reinforcement learning while being evaluated against standard scores.
6. Inputs. Input consists of text.
7. Model. The trained model includes millions of learned parameters.
8. Inference algorithm. GNMT is made accessible through an interface that everyone knows as Google Translate.
9. Outputs. Outputs consist of textual sentences in the target language.

Dataset risks

2. **Metadata:** Degrade generalization since a model learns a feature of the metadata instead of the content itself.
3. **Data rights:** Legally encumbered data are scraped from the internet for training, leading to legal entanglements.
4. **Partitioning:** Bad data partitions for training, validation and testing datasets may lead to a misbehaving ML system.
5. **Normalization:** Changes raw data, and may introduce too much bias.
6. **Annotation:** An attacker can introduce bias into the way data is annotated into features.
7. **Encoding integrity:** Pre-processing and encoding of the data can lead to integrity issues if the data has bias.
8. **Bad evaluation data:** A bad evaluation dataset can give unrealistic QA results before shipping to production.
9. **Storage:** Data may be stored and managed insecurely.
10. **Recursive pollution:** Bad model output into future training data can damage the accuracy and reliability of an ML model.

Continues on backside

Dataset risks

Cont.

J. Data integrity: Without proper integrity checks, distributed data can be tampered with undetected.
Q. Data confidentiality: Sensitive data that is built into the model through training can be disclosed by attacks or unintentionally.
K. Data Poisoning: An attacker manipulates data to influence ML training.
A. You have invented your own risk for machine learning datasets.

Output risks

2. **Cry wolf:** A security ML model raises too many alarms, its output is ignored.
3. **Black box discrimination:** ML systems operating based on personal data carry the risk of illegal discrimination.
4. **LLM overconfidence:** Dependence on an LLM without oversight may lead to issues like misinformation and legal concerns.
5. **Inscrutability:** Often, nbody iss sure how the trained systems do what they do. This negatively affects trustworthiness.
6. **Miscategorization:** Bad output due to internal bias, malicious input or other attacks may escape into the world.
7. **Transparency:** It is easier to perform attacks undetected on a black-box system which is not transparent.
8. **Confidence scores:** ML confidence scores exposes information useful to attackers.
9. **Wrongness:** LLMs can generate misinformation trying to predict the next tokens from a prompt.
10. **Excessive LLM agency:** The actions of an over-privileged LLM system may have severe consequences.

Continues on backside

Output risks

Cont.

J. Overconfidence: An ML model's output treated as high confidence data may cause unexpected issues.
Q. Error propagation: Errors in an ML subsystem may propagate in unforeseen ways to a larger system it's part of.
K. Output manipulation: An attacker gets between the ML and the receiver.
A. You have invented your own risk for machine learning output.

Input risks

2. **LLM feedback scores:** LLM chat systems allow user feedback as a parameter. This can be abused by attackers.
3. **Open to the public:** Makes an LLM model more susceptible to attacks from users.
4. **Sponge input:** Attacker provides an LLM system with input that is more costly to process.
5. **Input ambiguity:** Natural language is an ambiguous interface, making LLMs susceptible to misinformation.
6. **Text encoding:** An ML built for one text encoding can misperform if presented with a differently encoded text.
7. **Denial of service:** DoS attacks can have a massive impact on a critical ML system.
8. **User risk:** A user may expose their personal data to the ML owners.
9. **Dirty input:** Dirty inputs can be hard to process and may disrupt an ML system.
10. **Controlled input stream:** Outside sources of input may be manipulated by an attacker.

Continues on backside

Input risks

Cont.

J. Looped input: ML system output to the real world may feed back into training data or input, leading to a feedback loop.
Q. Prompt injection: Malicious text input to an LLM to trick it into misbehaving.
K. Malicious input: Input tricking the ML system into making a false prediction or categorization.
A. You have invented your own risk associated with machine learning input.

Model risks

2. **Catastrophic forgetting:** With too much overlapping information, the model starts “forgetting” information.
3. **Oscillation:** An ML system may end up oscillating and not converging if failing to use gradient descent.
4. **Randomness:** Setting weights and thresholds with a bad RNG can lead to subtle security issues.
5. **Online system manipulation:** Attackers can nudge the model so that it drifts from its intended operational profile.
6. **Overfitting:** The model learns its training data too well and is no longer able to generalize.
7. **Hyperparameters:** An attacker that controls the model’s hyperparameters can manipulate training.
8. **Hosting:** The server where the model is hosted is insufficiently protected.
9. **Hyperparameter sensitivity:** Badly configured sensitive hyperparameters can lead to overfitting.
10. **Model theft:** Attackers reverse engineer the model by observing input/output.

Continues on backside

Model risks

Cont.

J. Training set reveal: Data can potentially be extracted from the model.
Q. Trojanized model: Model transfer leads to the possibility that what is being reused may be a Trojaned version of the model
K. Improper re-use of model: The re-used model may be transferred into a problem space it's not designed for.
A. You have invented your own risk associated with machine learning models.