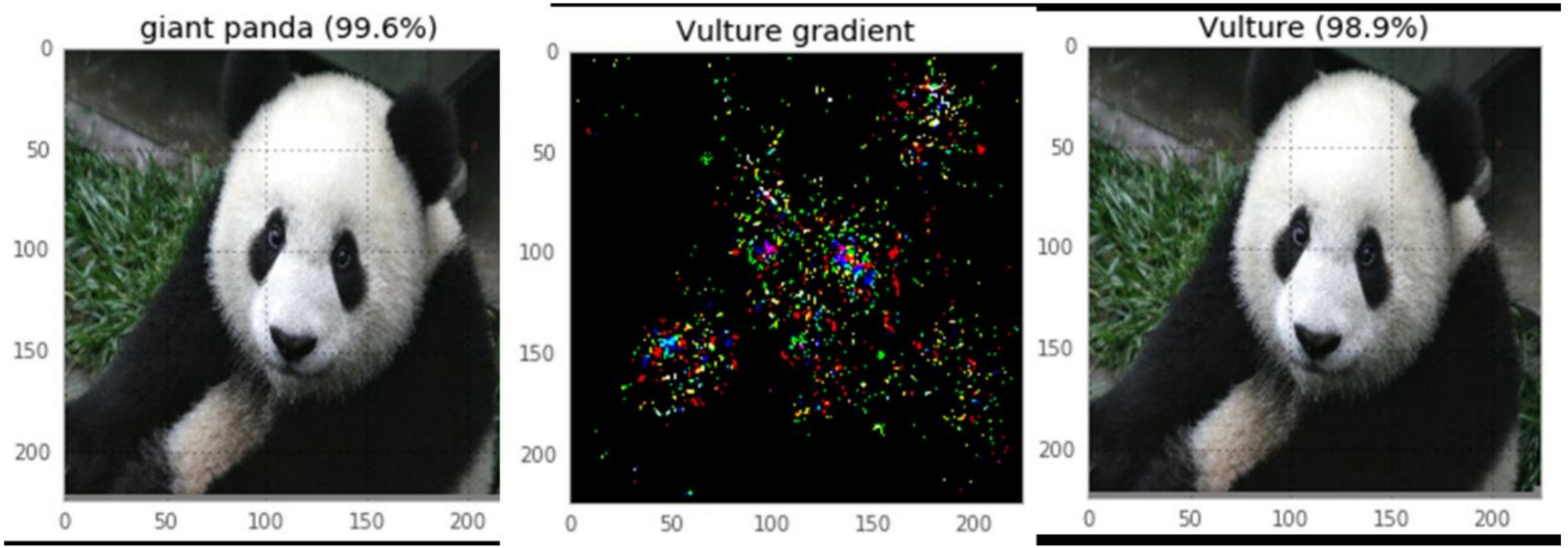
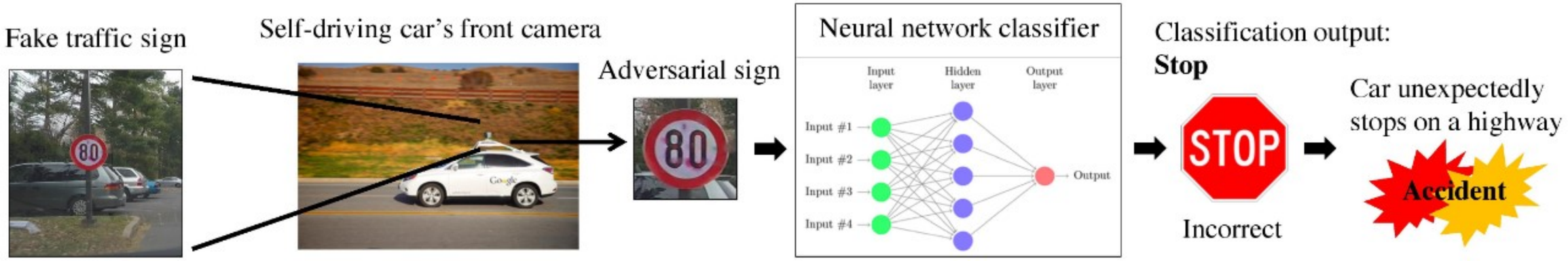
對抗式攻擊(adversarial attack)

透過在輸入資料中加上細微的擾動，使機器學習模型做出錯誤的判斷，
而這些擾動往往相當隱蔽、難以被肉眼察覺



對抗式攻擊(adversarial attack)

實例：自駕車影像辨識

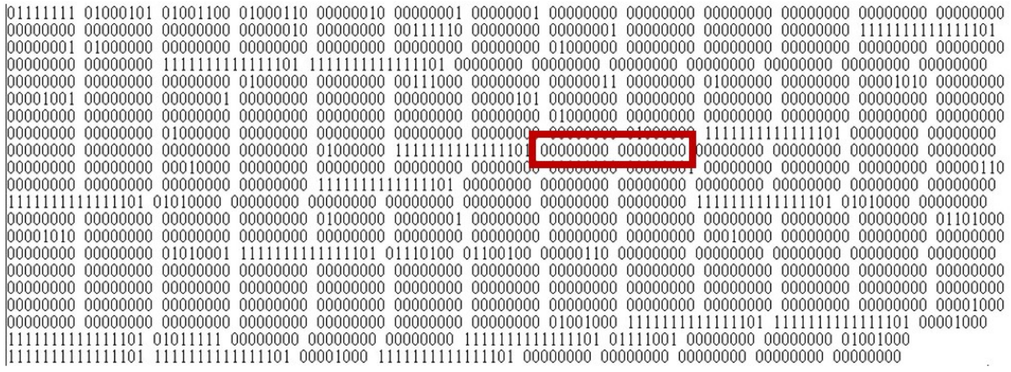


受到攻擊，可能會讓汽車失控，或造成更大的傷亡

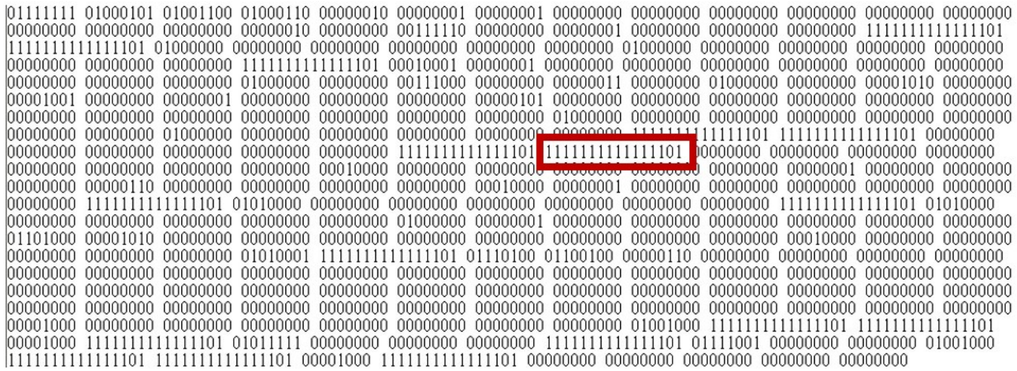
對抗式攻擊(adversarial attack)

實例：惡意程式躲避偵測

模型用來預測之binary file：



before attack



after attack

對抗式攻擊(adversarial attack)

實例：惡意程式躲避偵測

實際運行之程式碼：

```
push   %ebp
mov    %esp,%ebp
sub    $0x18,%esp
sub    $0xc,%esp
push   $0x80485b0
call   8048350 <printf@plt>
add    $0x10,%esp
sub    $0xc,%esp
lea    -0x18(%ebp),%eax
push   %eax
```

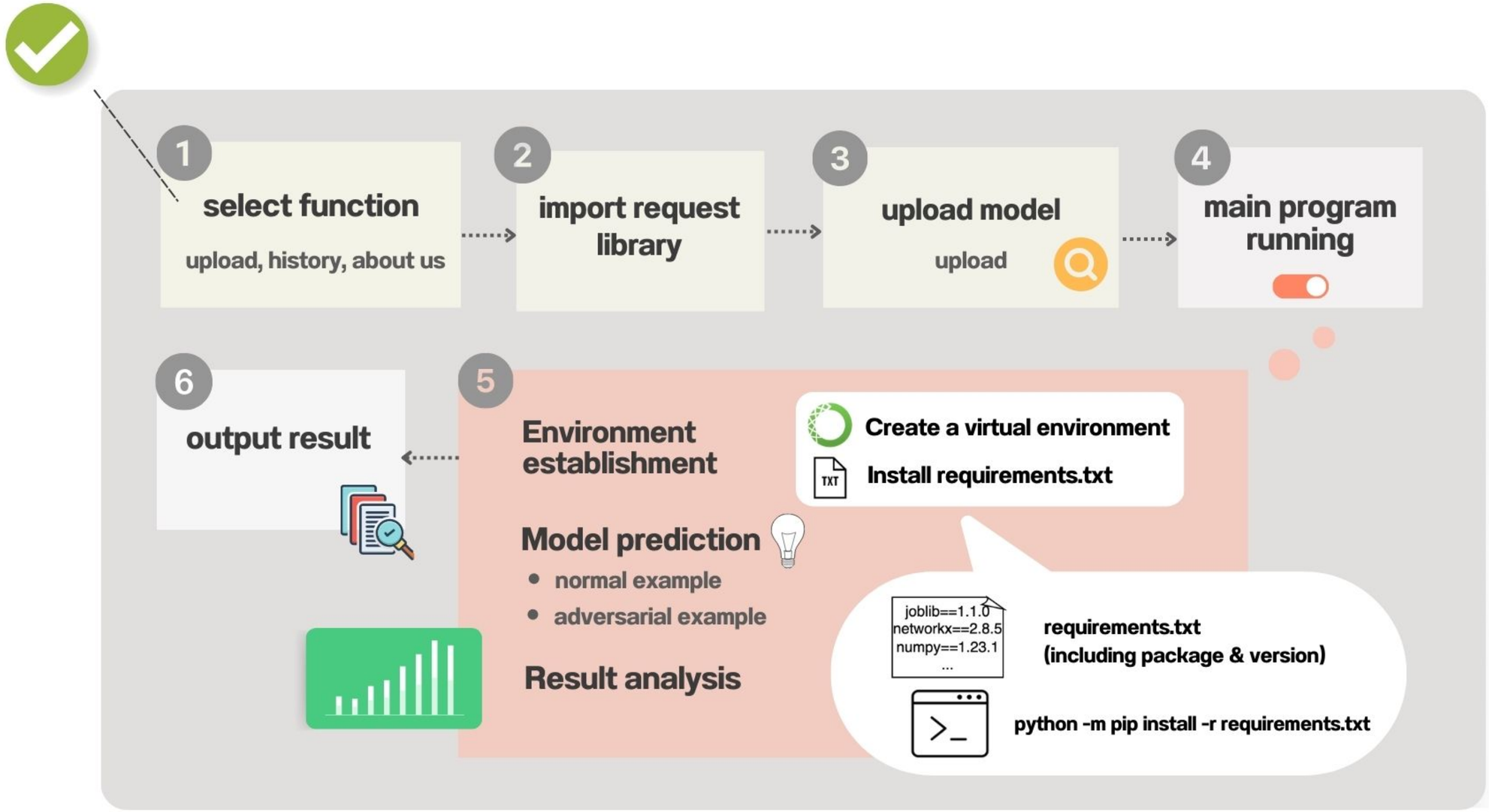
before attack

經過加上對抗式干擾後，雖被判定為良性軟體，但實際運行程式內容依舊不變，執行惡意程式

```
push   %ebp
mov    %esp,%ebp
sub    $0x18,%esp
sub    $0xc,%esp
push   $0x80485b0
call   8048350 <printf@plt>
add    $0x10,%esp
sub    $0xc,%esp
lea    -0x18(%ebp),%eax
push   %eax
```

after attack

系統流程


1
select function
upload, history, about us
2
import request library
3
upload model
upload
4
main program running
6
output result
5
Environment establishment
Create a virtual environment
Install requirements.txt
Model prediction
normal example
adversarial example
Result analysis
joblib==1.1.0
networkx==2.8.5
numpy==1.23.1
...
requirements.txt
(including package & version)
python -m pip install -r requirements.txt

結語

AI模型預測即使能便利我們的生活，其安全性卻是一大難題，若無法事先檢測出潛在問題，並對未來可能遇到的錯誤進行預期性排除，其結果將會不堪設想