# 摘要

我们建议使用降维来防御针对 ML 分类器的规避攻击。我们研究了一种通过主成分分析来降维的策略，以增强机器学习的适应能力，既可以应用于分类又可应用于训练阶段。我们使用多个真实世界的数据集证明了数据降维在防御逃避攻击方面的可行性。我们的主要研究结果是：（1）有效对抗文献中的策略性的逃避攻击，将对方成功攻击所需的资源增加约 2 倍，（2）适用于一系列 ML 分类器包括支持向量机和深度神经网络，（3）可推广到多个应用领域，包括图像分类和人类活动分类。


# 1 介绍

我们生活在一个到处充满着机器学习（ML）和人工智能的时代。机器学习被用于诸如图像识别，自然语言处理，垃圾邮件检测，车辆自动驾驶甚至恶意软件检测等多种基础应用中。

此外，最近在深度学习方面取得的进展表明分类的准确性可以接近于人类操作的准确性，这使得 ML 系统的广泛应用成为可能。鉴于 ML 应用程序的无处不在，它越来越多地应用于敌对情景中，在这种情况下，攻击者可以从 ML 系统的失败中对输入进行正确的分类。那么问题就出现了：ML 系统在对抗环境中安全吗?

对抗性机器学习：从 21 世纪初开始，已经有大量工作将机器学习算法的脆弱性暴露给战略对手。例如，中毒攻击在训练阶段系统地引入敌对数据，从而在测试阶段导致数据分类错误。另一方面，规避攻击的目的是通过向测试数据中添加策略性的干扰数据来欺骗现有的 ML 分类器。

规避攻击：在本文中，我们重点关注规避攻击，其中攻击者的目标是干扰 ML 分类器的测试输入以引起错误分类。针对各种机器学习分类器都提出过规避攻击，如支持向量机，基于树的分类器，随机森林和增强树，以及最近的神经网络。使用机器学习的应用程序（例如人脸检测，语音命令识别和 PDF 恶意软件检测）的脆弱性也已得到证明，这也突出了防御的必

要性。令人惊讶的是，这也表明，敌对方修改后的数据（针对特定分类器）的规避属性持续存在于不同的 ML 分类器中，这使得即使对 ML 系统了解很有限的对手都可以攻击它。因此，在敌对情境下使用 ML 系统时考虑敌对数据和躲避攻击的可能性至关重要。然而，针对这些攻击的防御措施极少，并且每种攻击的适用性仅限于某些已知的攻击和特定类型的 ML 分类器（请参见第 7 节获得详细描述）。

## 1.1 贡献

通过广泛的评估，我们发现我们的防御机制明显降低了逃避攻击的成功率。就我们所知，这是针对具有以下属性的规避攻击的唯一防御措施：（1）适用于多个 ML 分类器（如 SVM，DNN），（2）适用于多个应用领域（图像和活动分类），（3）减轻多种攻击类型，包括战略攻击类型。此外，我们的防御可调性允许系统设计人员根据应用选择公共安防权衡曲线上适当的操作点。

### 1.1.1 防御

在本文中，我们提出使用数据的降维来防御针对 ML 系统的规避攻击。降维技术（如主成分分析）旨在将高维数据投影到较低维度的空间，同时满足特定的条件。我们研究了一种降维的策略，以增强机器学习的适应能力，既可以应用于分类又可应用于训练阶段。我们考虑一种方法，将降维应用于训练数据和测试数据，以增强训练分类器的可靠性。

### 1.1.2 实证评估

我们证明了我们的防御措施的可行性和有效性：

- 多重分类器，例如支持向量机（SVM）和深度神经网络（DNN）
- 几种不同类型的规避攻击，例如 Moosavi-Dezfooli 等人对线性 SVMs 的攻击、Goodfel-low 等人的深层神经网络攻击以及针对我们的防御

措施的策略性攻击

- 各种现实世界的数据集/应用程序：MNIST 图像数据集和 UCI 人类活动识别（HAR）数据集。

我们的主要发现是，即使面对一个几乎完全了解 ML 系统的强大对手，(1) 我们的防御措施使得成功攻击所需的修改程度有着高达 5 倍的显著提高，同样的，以固定的修改程度攻击的成功率降低约 2-50 倍，(2)防御措施可以用于不同的 ML 分类器，对原始分类器进行最小限度的修改，同时仍然有效地防御攻击，（3）在大多数情况下良性样品的分类成功率有约 1-4 % 的适度变化。我们还提供了公共安防权衡曲线的分析以及我们的防御措施产生的计算开销。我们的结果开源在https://github.com/inspire-group/ml_defense上。

然而，我们的防御措施并没有完全解决规避攻击的问题，因为它可以降低固定预算下的敌对成功率，但这并不是在所有情况下都忽略不计。在第 4 节中，我们讨论了对手在不同应用场景下可用的预算范围，并明确了防御有效的场景。我们希望我们的工作能够激发进一步的研究，以解决规避攻击来保证机器学习的系统的安全性。

本文的其余部分安排如下：首先，在第 2 节中，我们介绍了对抗机器学习的必要背景。然后，在第 3 节中，我们描述了我们的防守措施。接下来，我们分别在第 4 节和第 5 节中设置并提出我们的实证评估。我们在第 6 节讨论我们的结果。最后，我们在第 7 节中详细介绍相关工作，并在第 8 节中做出结论。

## 2 对抗性机器学习

在本节中，我们提出了对抗性机器学习所需的背景，重点关注（a）ML 分类器，如 SVM 和 DNN，以及（b）通过干扰测试输入引发错误分类的规避攻击。

动机和运行示例：我们的运行示例使用来自 MNIST 数据集的图像数

据（详见第 4 节）。图 1（a）描绘了来自 MNIST 数据集的正确测试图像，这些图像被 SVM 分类器正确分类；而图 1（b）描绘了对手制作的测试图像（使用 Papernot 的规避攻击的扰动图像），它们被 SVM 分类器错误分类。

(a) 来自 MNIST 数据集的典型测试图像。正确分类为 7,2,1,0 和 4。

(b) 使用对线性 SVMs 的规避攻击获得的相应敌对图像 [29]。分别错误地分类为 9,9,3,2 和 0。

图 1：从 MNIST 数据集取得的良性和对抗性图像的比较。

## 2.1 使用机器学习分类

在本文中，我们关注有监督的机器学习，其中分类器通过预先存在的标签对数据进行训练。一个训练完成的监督机器学习分类器是一个函数，通过输入点 $\boldsymbol{x} \in \mathbb{R}^d$（二进制时为 $\{0,1\}d$），会输出 $\hat{y} \in C$，其中 C 是所有可能分类的集合。例如，在 MNIST 数据集的情况下，$\boldsymbol{x}$ 将是 28×28 像素的手写数字的灰度图像，而 C 将是有限集合 {0,1,2,3,4,5,6,7,8,9}。

## 2.2 攻击机器学习系统

在本小节中，我们首先讨论对抗模型，之后我们会讲述一般的规避攻击，最后讲述对特定的 ML 分类器的规避攻击。

注: 我们把完整的训练集表示为 $S_{train}$, 完整的训练数据表示为 $S_{test}$, 将 ML 分类器表示为 $f$, 并且针对 ML 分类器的特定参数表示为 $\boldsymbol{\theta}$。数据的原始维度表示为 $d$。接下来, 我们把攻击者的攻击算法表示为 $A(\boldsymbol{x}_{in}|K)$, 其中, $\boldsymbol{x}_{in}$ 表示对手开始时的输入, $K$ 代表对手的已知信息, 可能是 $\{S_{train}, f, \boldsymbol{\theta}\}$ 的任一子集。$\widetilde{\boldsymbol{x}}$ 表示 A 生成的敌对样本。

### 2.2.1 敌对方的目标和能力

在本文中, 我们关注的情景是, 攻击者的目标是通过修改一个正确的输入, 以便使它被误分为其他的任何分类, 或者使其被归类为与原始类不同的目标分类。请注意, 这些目标分类在二元分类器的情况下是等价的。

我们的基本假设是对手具有以下能力。

- 对手完全了解原始分类器已经训练过的训练集, 即她知道分类器作为输入所采用的特征向量的类型。
- 对手知道分类器结构, 超参数和训练过程。
- 错误数据是由对手离线创建的, 在测试阶段提交给 ML 分类器。

简而言之, $\widetilde{\boldsymbol{x}} = A(\boldsymbol{x}_{in}|S_{train}, f, \boldsymbol{\theta}, K_{add})$, 其中 $K_{add}$ 表示关于对手可能拥有的系统的任何其他知识。

我们对对手的能力的假设是保守的, 因为从安全角度来讲, 系统在完全了解系统安全的对手的强力的攻击下, 依旧是健壮的。而且, 一个有着 ML 系统知识的攻击者, 即使有着有限权限的访问 (如黑盒访问), 也可以很好的对分类器进行推断来进行规避攻击。这和一个拥有完全访问权的对手攻击的效果集合一样, 这证明了我们的假设是合理的。

### 2.2.2 规避攻击

在正常操作, 即没有攻击者时, 当输入 $\boldsymbol{x}_i \in S$, $f$ 会输出 $\hat{y}$, 其中 $S$ 是输入集合。输出的分类中正确匹配的比例为 $\alpha$, 即,

$$\alpha(S) = \frac{\#\{(\boldsymbol{x}, y) \in S : f(\boldsymbol{x}) = y\}}{\#S} \tag{1}$$

其中　给出了一组的基数。攻击者的目标是设计一个作用在 $x \in S$ 上的算法 A 来生成敌对数据，即，$A(\boldsymbol{x}) = \widetilde{\boldsymbol{x}}$，令

$$S^{adv} = \{(A(\boldsymbol{x}), y) : (\boldsymbol{x}, y) \in S\}$$

这是一组对比修改的例子，其中修改之处应满足：

- 与分类器的正常操作相比，增加错误分类的占比，即 $\alpha(S^{adv}) < \alpha(S)$，
- 在诸如图像和文本等人类可解释的数据的情况下，不被人类察觉到异常；在诸如恶意软件样本，网络和系统日志等数据的情况下，可被基于规则的检测系统通过。例如：在恶意软件的情况下，攻击者受到这样的限制，即她的修改必须确保最终的样本仍然是恶意的。

  我们接下来讨论敌对干扰，以及在图片数据的情况下，他们对人类的感知力。

### 2.2.3　敌对干扰

模拟人类对图像扰动的感知是一个难题。作为人类可感知性的代理，我们将对某个范数 $||\cdot||$ 的修正程度定义为 $||A(\boldsymbol{x}) - \boldsymbol{x}||$。需要强调的是，我们将考虑受 $\ell_2$ 约束的限制，即 $||\widetilde{\boldsymbol{x}} - \boldsymbol{x}||_2 \leqslant \xi$，其中 $\xi$ 决定了干扰的强度。[35] 中给出了用于约束敌对干扰的各种规范与其感知之间关系的详细描述。

现在，我们定义了实现不同对抗目标所需的最小扰动。为了在特定的类 $z$ 中导致错误分类，必须添加一个输入数据 $(\boldsymbol{x}, y)$ 作为最小的扰动，其中 $z \neq y$，

$$\Delta(\boldsymbol{x}, z) = \inf_{\widetilde{\boldsymbol{x}}}\{||\widetilde{\boldsymbol{x}} - \boldsymbol{x}|| : f(\widetilde{\boldsymbol{x}} = z)\}$$

这是导致 $\boldsymbol{x}$ 被归类为 $z$ 所需的最小失真。导致 $\boldsymbol{x}$ 在任何类中被错误分类所需的最小失真是，

$$\Delta(x) = \min_{z \in C \setminus \{y\}} \Delta(\boldsymbol{x}, z)$$

对于图像数据，这些量与最小可检测失真之间的关系决定了分类器 $f$ 对敌对扰动的鲁棒性。在图 1 中，图像中的干扰值导致线性 SVM 几乎将所

有输入都错误错误，但干扰对于人眼几乎不可见。这表明线性标准形式的 SVM 对抗扰动是不稳健的。在第 4.4 节中进一步讨论了用于约束对手的指标。

## 2.3 针对特定分类器的规避攻击

我们现在描述现有文献记载的针对特定 ML 分类器的攻击，并展示来自 MNIST 数据集的一些对抗性例子。表 1 给出了各种攻击的总结。

### 2.3.1 对线性 SVM 的最佳攻击

在线性支持向量机的多类分类设置中，分类器 $g_i$ 针对每个类别 $i \in C$ 进行训练，其中

$$g_i : \boldsymbol{x} \mapsto \boldsymbol{w}_i^T \boldsymbol{x} + b_i \tag{2}$$

$\boldsymbol{x}$ 被分配给类 $f(\boldsymbol{x}) = \arg\max_{i \in C} g_i(\boldsymbol{x})$。假定真正的类别是 $t \in C$，攻击的目标是找到最接近的点 $\widetilde{\boldsymbol{x}}$，使得 $f(\widetilde{\boldsymbol{x}}) \neq t$。

从 [29] 我们知道，对于多类分类器的最优无目标的攻击，即如果我们只关心 $f(\widetilde{\boldsymbol{x}})$ 使得 $||\widetilde{\boldsymbol{x}} - \boldsymbol{x}||$ 尽可能小，令 $\widetilde{\boldsymbol{x}}$ 的最优选择是 $\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}$，则，

$$k = \arg\min_j \frac{g_t(\boldsymbol{x}) - g_j(\boldsymbol{x})}{||\boldsymbol{w}_t - \boldsymbol{w}_j||} \tag{3}$$

进而得到，

$$\widetilde{\boldsymbol{x}}(\xi) = \boldsymbol{x} + \xi \frac{\boldsymbol{w}_t - \boldsymbol{w}_k}{||\boldsymbol{w}_t - \boldsymbol{w}_k||} \tag{4}$$

这里的 $\xi$ 代表干扰的程度。导致误分类的 $\xi$ 的最小值是 $\xi^* = \frac{|g_t(\boldsymbol{x} - g_k(\boldsymbol{x}))|}{||\boldsymbol{w}_t - \boldsymbol{w}_k||}$。很容易可以证明 $f(\widetilde{\boldsymbol{x}}(\xi^*)) = k$。请注意，由于 $||\xi \frac{\boldsymbol{w}_t - \boldsymbol{w}_k}{||\boldsymbol{w}_t - \boldsymbol{w}_k||}|| = \xi$，这个攻击受到 $\ell_2$ 约束的限制。

### 2.3.2 基于梯度的神经网络攻击

FGS 攻击是 [19] 中引入的针对神经网络的高效攻击。在这种情况下，通过添加与损失函数的梯度（$\nabla J_f(\boldsymbol{x}, y, \theta)$）成正比的对立噪声来生成对抗

示例，其中 $J_f(\cdot)$ 表示损失函数，$\theta$ 表示用于训练的超参数。可以使用反向传播有效地计算梯度。具体为，

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \eta sign(\nabla J_f(\boldsymbol{x}, y, \theta)) \tag{5}$$

其中 $\eta$ 是对手可以改变的参数，以控制对抗性例子的有效性。随着 $\eta$ 的增加，攻击的成功率一般也在增长。然而，较大的干扰可能会使人们难以辨识图像（请参阅附录中的图像，其变化范围为 $\eta$）。FGS 攻击者受到 $\ell_2$ 约束的限制，因为 $||\eta sign(\nabla J_f(\boldsymbol{x}, y, \theta))||_\infty = \max_i |\eta sign(\nabla J_f(\boldsymbol{x}, y, \theta))_i| = \eta$，这控制着干扰的程度。

FGS 攻击和对线性 SVM 的攻击根据不同的标准受到限制。为了便于比较各种分类器的鲁棒性以及我们对它们的防御效果，我们提出了一种修改 FGS 攻击的方法，该攻击受到 $\ell_2$ 约束的限制。我们将这称为快速梯度（FG）攻击，我们将敌对示例定义为，

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \eta \frac{\nabla J_f(\boldsymbol{x}, y, \theta)}{||\nabla J_f(\boldsymbol{x}, y, \theta)||} \tag{6}$$

对于 FG 攻击而言，$\eta$ 是对干扰标准 $\ell_2$ 的约束。

## 2.4 降低机器学习的维度

在处理高维数据（意味着每个样本具有大量特征）的同时，很难弄清哪些特征很重要。应用程序约束也可能会使原始高维空间中的数据执行学习任务变得不切实际。因此降维是对高维数据有效的预处理步骤。它还可以帮助解决与“维度诅咒”相关的问题。在这种情况下，数据首先投影到较低维空间，然后将其作为 ML 系统的输入。常见的降维算法是 PCA [27]，随机投影 [36] 和核 PCA [37]。在本文中，我们使用 PCA 等降维方法不仅有助于解释性和提高效率，而且还可以提高 ML 系统对敌对实例的鲁棒性。

表 1: 对线性 SVM 和神经网络的攻击总结

| 攻击 | 分类器 | 约束 | 直观结果 |
|---|---|---|---|
| 线性 SVM 最优攻击 | 线性 SVM | $\ell_2$ | 趋向分类器边界 |
| 快速梯度 | 神经网络 | $\ell_2$ | 最小扰动方向的一阶近似 |
| 快速渐变标志 | 神经网络 | $\ell_\infty$ | 建议不断缩放每个像素模型 |

# 3 基于降维的防御

在前面的章节中，我们已经看到许多 ML 分类器容易受到各种不同的规避攻击。这些漏洞使得对手很容易制作输入以达到他们期望的目标，这可能包括欺骗自动驾驶汽车 [4,5]，并使其崩溃为逃避欺诈和恶意软件检测 [6,7] 。显然需要 ML 系统的防御机制来针对各种攻击，因为在之前，系统的所有者不知道针对系统的可能攻击范围。此外，找到适用于各种分类器的防御措施，可以引导我们更好地理解 ML 系统首先易受攻击的原因。在本节中，我们提出至少对文献中常见攻击是不可知的防御，即使存在一个更强大的对手的情况下，它仍然是有效的。此外，防御使得在不同应用场景下运行的多类 ML 类更加健壮，如第 5 节所示。防御是基于降维，这在第 2 节中已经简要介绍。

现在，我们介绍我们的威胁模型和设计目标，然后介绍我们的基于降维的防御机制。在下文中，我们将原始的 ML 分类系统称为“强大的分类管道”，并增加了防御机制。

## 3.1 威胁模型

鉴于 2.2 节定义的对抗能力，我们在评估防御机制的有效性时考虑了威胁模型中的两类逃避攻击：

1 Vanilla 攻击：这个类别考虑了来自文献的一系列现有攻击，没有任何

修改。由于这些攻击是针对机器学习系统设计的，而没有考虑到任何防御措施，对手不了解防御机制。我们注意到，防御机制的设计可以成功地缓解甚至是已知的攻击，这是安全社区面临的重大挑战。

2 战略攻击：在第二种情况下，对手知道防御机制。这意味着对手在制作敌对的实例时会考虑到防御的效果，甚至可能转而采用针对特定国防使用而优化的不同攻击策略。

为了彻底评估我们的防御，我们调查了第二种新的攻击策略，为我们的防御进行了优化。我们表明，即使在这种情况下，我们的防御措施也是有效的，尽管程度较低。

## 3.2 设计目标

任何有监督的机器学习分类器的主要目标是在测试集上实现最佳的准确性。此外，希望机器学习分类器尽可能高效。有鉴于此，我们必须确保任何防御机制对整体分类过程的效率或准确性都没有太大的影响。因此，防御机制的设计目标是：

1 高分类准确度：将防御机制添加到整个分类流水线应该保持良性测试集的高分类准确性。实际上，一个理想的防御机制甚至可以通过减少过度配合来增加测试集的分类准确性，增强分类器的能力等。

2 高效率：时间和空间的复杂性决定了算法的效率。在最坏的情况下，防御机制不应该在运行时间和原始分类器所需的空间上增加多项式开销。在理想的情况下，防御机制会使整个管道在时间和空间上更高效。

3 安全性：我们将机器学习分类器的脆弱性定义为被错误分类的对手修改过的输入数据的分数（受限于原始输入数据被正确分类 2）。为了使防御有效，我们需要这种错误分类比原分类器更低。因此，对于一个特定的攻击，我们可以说，如果对分类的例子错误分类的比例小于最初的分类器，那么对于一个可比较的攻击参数，流水线更安全。

4 可调性：根据应用，性能（即分类准确性），效率或安全性可能是机器学习系统用户首要关注的问题。防御机制应该让用户通过修改其参数来在性能，效用和安全性的多维权衡空间中导航不同点。

在第 5 部分中，我们量化了我们的防守如何满足上述设计目标。现在，我们提供强大的分类管道的概述，以及它如何帮助抵御现有攻击。

## 3.3 防御概述

在我们的防御中，我们利用降维技术将高维数据投影到较低维空间，并通过修改训练阶段使分类器更具有可重用性。在我们的防御的第一步中，ML 系统用户选择降低的维度 $k < d$。然后，使用以 $k$ 和 $\boldsymbol{X}^{train}$ 作为输入的降维算法 DR，将训练数据 $\boldsymbol{X}^{train}$ 转换为较低维空间。然后在降维训练集上训练新的分类器 $f_k$。在测试时间，使用相同的算法 DR 将所有输入变换到较低维空间，并且将减小的维度输入 xk 作为输入直接提供给 fk。我们现在使用 PCA 描述防御的具体实例。

## 3.4 使用 PCA 防御

### 3.4.1 实施防御

在我们的使用 PCA 的防御中，主要组成部分是使用适当的以平均值为中心并且按比例缩放的训练数据矩阵来计算的。如果有 n 个训练样本，则用于计算主成分的矩阵是 $\boldsymbol{X}^{train}$，它是一个 $d \times n$ 的矩阵。所有后续数据样本都被投影到最初计算的这些主成分上。使用算法 1 生成弹性降维分类器 $f_k$。它采用分类器训练算法训练并对其进行调整以生成使用降维的变体。算法 1 的输入是：

- $f_k^0$：这是使用适当参数初始化的初始未训练分类器。
- $\boldsymbol{\theta_k}$：这些是 $f_k$ 训练中使用的与 $\boldsymbol{\theta}$ 相同的参数，除了由于降维训练可能会改变的任何尺寸特定参数。

算法 1 DR Train

输入：$f_k^0, \boldsymbol{\theta}_k, k, \boldsymbol{X}^{train}, Train$
输出：$f_k$
1: 使用 PCA 找到矩阵 $U_k$ 的前 k 个主要元素来给 $\boldsymbol{X}^{train}$ 降维
2: 计算降维训练集 $\boldsymbol{U}_k^T \boldsymbol{X}^{train}$
3: 训练降维分类器 $f'_k = Train(f_k^0, \boldsymbol{\theta}_k, \boldsymbol{U}_k^T \boldsymbol{X}^{train})$
4: 令 $f_k = \boldsymbol{x} \longmapsto f'_k(\boldsymbol{U}_k \boldsymbol{x})$

算法 2 替代 DR Train

输入：$f_k^0, \boldsymbol{\theta}_k, k, \boldsymbol{X}^{train}, Train$
输出：$f_k$
1: 使用 PCA 找到矩阵 $U_k$ 的前 k 个主要元素来给 $\boldsymbol{X}^{train}$ 降维
2: 计算项目的训练集 $\boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{X}^{train}$
3: 令 $f_k = Train(f_k^0, \boldsymbol{\theta}_k, \boldsymbol{U}_k \boldsymbol{U}_k^T, \boldsymbol{X}^{train})$

- $k$：这是降维需要的参数。基于安全性的权衡 $k$ 可以被适当改变。
- Train：这是用于训练所需分类器的算法。一个常用的训练算法的例子是随机梯度下降 [38]。

对于 Train 的某些选择，算法 1 和 2 是等价的，即在给定相同输入的情况下它们将输出相同的分类器。其中一个重要例子是使用 $\ell$ 正则化训练的线性 SVMs。在这种情况下，算法 1 和 2 将清楚地选择在每个训练点上具有相同行为的线性函数。这通过线性延伸到包含所有投影数据点的 $\mathbb{R}$ 的子空间。

在实现这种行为的线性函数中，正则化会迫使算法 2 选择 $\ell$ 范数中具有最小权重向量的函数。这将是与包含投影数据的子空间正交的所有矢量不变的线性函数。实际上，我们更喜欢算法 1，因为它在向量的较低维度上

运行，因此计算效率更高。

### 3.4.2 PCA 简介

PCA [27] 是数据的线性变换，在数据空间中统一所谓的“主轴”，这是数据具有最大变化的方向，并沿这些轴投影原始数据。选择沿着 k 个主轴投影数据可以减少数据的维数。$k$ 的选择取决于保留原始差异的百分比。直观上，PCA 识别“信号”或数据中有用信息的方向，并将其余部分作为噪声丢弃。

具体来说，令数据样本为 $\boldsymbol{x}_i \in \mathbb{R}(i \in \{1, ..., n\})$，令 $\boldsymbol{X'}$ 为一个 $dxn$ 的矩阵，$\boldsymbol{X'}$ 的每一列对应一个样本，并且令 $\mathbf{1} \in \mathbb{R}^n$ 表示为每个向量。于是，$\frac{1}{n}\boldsymbol{X'}\mathbf{1}$ 代表一个样本，并且 $\boldsymbol{X} = \boldsymbol{X'}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$ 代表中心样本的矩阵。$\boldsymbol{X}$ 的主成分是其样本协方差矩阵 $\boldsymbol{C} = \boldsymbol{X}^T\boldsymbol{X}$ 的归一化特征向量。更准确地说，因为 $\boldsymbol{C}$ 是正半定，所以存在 $\boldsymbol{C} = \boldsymbol{U} \wedge \boldsymbol{U}^T$，其中 $\boldsymbol{U}$ 是正交的，$\wedge = diag(\lambda_1, ..., \lambda_n)$，并且 $\lambda_1 \geq ... \geq \lambda_n \geq 0$。请注意，$\boldsymbol{U}$ 是其列是 $\boldsymbol{C}$ 的单位特征向量的 $d \times d$ 的矩阵，即，$\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_d]$，其中 $\boldsymbol{u}_i$ 是 $\boldsymbol{C}$ 的归一化特征向量。特征值 $\lambda_i$ 是沿着第 $i$ 个主分量的 $\boldsymbol{X}$ 的方差。

$\boldsymbol{U}^T\boldsymbol{X}$ 的每一列都是以主成分为基础的数据样本。令 $\boldsymbol{X}_k$ 为样本数据在由 $k$ 个最大主成分跨越的 k 维子空间中的投影。因此 $\boldsymbol{X}_k = \boldsymbol{U}\boldsymbol{I}_k\boldsymbol{U}^T\boldsymbol{X} = \boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{X}$，其中 $\boldsymbol{I}_k$ 是对角线矩阵，其中前 k 个位置中的一个和最后一个 d-k 中的零，并且 $Uk = [\boldsymbol{u}_1, ..., \boldsymbol{u}_k]$。保留的方差量是 $\sum_{i=1}^{k} \lambda_i$，这是 k 个最大特征值的总和。

## 3.5 防御预期

现在，我们将对维度降维如何提高支持向量机的弹性提出一些预测。我们为了简单而讨论了两类案例，但是这些思想推广到了多类案例。

线性分类器的核心是函数 $g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$。$\boldsymbol{x}$ 和 $\boldsymbol{w}$ 都可以以主成分为基础来表示为 $\boldsymbol{U}^T x$ 和 $\boldsymbol{U}^T\boldsymbol{w}$。我们期望 $\boldsymbol{U}^T\boldsymbol{w}$ 的许多最大系数对应于 $\boldsymbol{C} = \boldsymbol{U} \wedge \boldsymbol{U}^T$ 的最小特征值。

原因很简单：为了使不同的主成分在分类器输出上达到相同的作用，$(\boldsymbol{U}^T\boldsymbol{w})_i$ 的系数必须与 $1/\sqrt{\lambda_i}$ 成正比。为了利用具有小变化的分量的信息，分类器必须使用大系数。当然，主要组成部分对于分类并不相同。然而，在有用的组成部分中，我们预计随着 $\sqrt{\lambda_i}$ 的增加，$(\boldsymbol{U}^T\boldsymbol{w})_i$ 系数也在减小。

图 2 中最上面的图表验证了这个预测。第一个主成分 3（每个图表的右上方）是迄今为止最有用的分类信息来源。因此它不符合整体趋势并且实际上具有最大的系数。然而，这一趋势仍然存在。

此外，我们预期较高的方差分量包含比较低方差分量更好的信息。通过更好的信息，我们指的是从训练集更好地推广到测试集的组件，或等价的组件，这些组件大多对应于类的下层特征，而不是由特定训练示例驱动的虚假关系。

由于线性分类器的最优攻击扰动是 w 的倍数，因此具有较大系数的主要分量是攻击者有利的主分量。投射防御否认了攻击者的这个机会。通过删除较低组件的所有变化，它会导致分类器不为其分配权重。这显着改变了分类器学习的结果 w。分类器失去了对某些信息的访问权限，但访问该信息需要大量权重系数，因此攻击者受到的伤害更大。

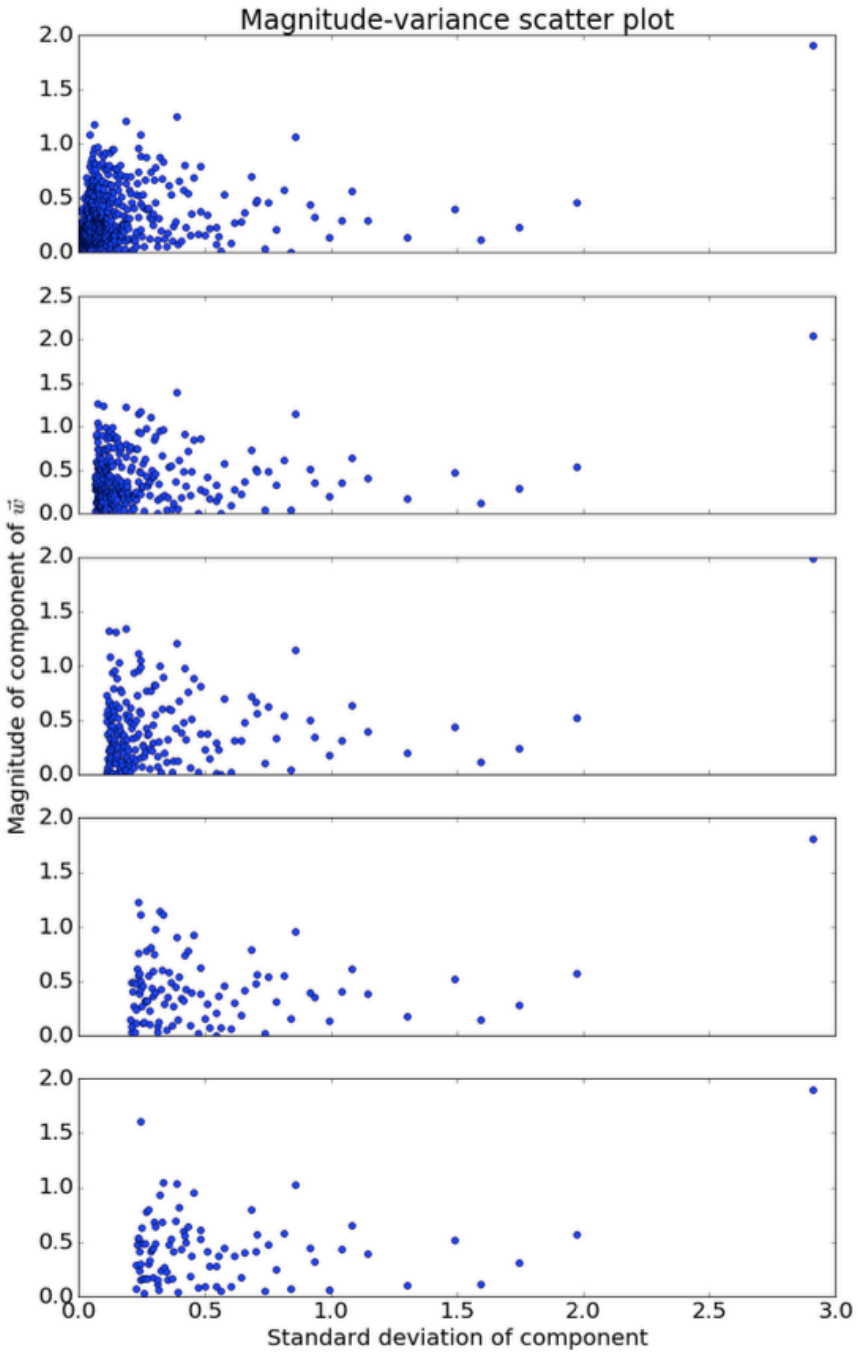


图 2：以主成分为基础的权重向量 w 的系数。横轴表示 $\sqrt{\lambda_i}$，纵轴表示

$|(\boldsymbol{U}^T\boldsymbol{w})_i|$。最上面的图是来自用原始 MNIST 数据训练的分类器，没有投影（即 $k = 728$）。接下来的四个图是用 $k \in \{300, 100, 50, 40\}$ 对投影数据进行训练的分类器。

### 3.6 防御的复杂性分析

使用 PCA 的防御增加了一次性的 $\Theta(d^2n + d^3)$ 开销以找到主成分，第一项来自协方差矩阵计算，第二项来自特征向量分解。在降维数据上训练一个新的分类器也有一次性开销。由于输入数据的维数减少，训练新分类器所需的时间将少于原分类器所需的时间。由于投影到主要组件上所需的矩阵乘法，每个后续输入都会产生 $\Theta(dk)$ 开销。

## 4 进行实验

在本节中，我们将提供数据集的简要说明和实施细节，机器学习和降维算法以及我们实验中使用的度量。我们还讨论了选择各种攻击的敌对预算的原因，并在 MNIST 数据集的情况下用可视的数据证实它们。

### 4.1 数据集

在我们的评估中，我们使用两个数据集。第一个是 MNIST 图像数据集，第二个是 UCI 人类活动识别数据集。我们现在分别进行详细介绍。

MNIST 数据集：这是手写数字图像的数据集 [30]。有 60000 个训练数据和 10000 个测试例子。每幅图像属于从 0 到 9 的单类。图像的尺寸为 28×28 像素（共 784 个）并且是灰度的。数字是大小是标准化和居中的。此数据集通常用作最先进分类器的“健全性检查”或第一级基准。我们使用这个数据集，因为它已经从以前的工作攻击视角进行了广泛的研究。我们很容易看到我们的防御对这个数据集的影响。

使用智能手机数据集的 UCI 人类活动识别（HAR）：这是从智能手机的加速度计和陀螺仪获得的测量数据集 [31]，而参与者则执行六个活动中

的一个。在 30 名参与者中，有 21 人被选来提供训练数据，其余 9 人则是测试数据。有 7352 个训练样本和 2947 个测试样本。每个样本具有 561 个特征，这些特征是从加速度计和陀螺仪获得的各种信号。这六类活动包括散步，上楼散步，下楼散步，坐着，站立和铺设。我们使用这个数据集来证明我们的防御工作适用于各种数据集和应用程序。

## 4.2 机器学习算法

我们已经对多种机器学习算法进行了评估,包括线性支持向量机(SVM)和各种具有不同配置的神经网络。所有实验都运行在运行 Ubuntu 14.04 的桌面上，运行频率为 4.00GHz 的 4 核 Intel R CoreTM i7-6700K CPU，24 GB 内存和 NVIDIA R GeForce R GTX 960 GPU。

线性支持向量机：简单的训练和分离超平面权重的可解释性使得线性支持向量机在广泛的应用中得到了应用 [37,39]。我们使用 Python 包 scikit-learn[40] 中的 LinearSVC 实现来进行我们的实验，默认情况下使用'one-versus-rest' 方法进行多类分类。

在我们的实验中，我们获得 MNIST 数据集的分类准确率为 91.5%，HAR 数据集的分类准确率为 96.7‰

神经网络：神经网络可以通过改变层数，神经元的激活函数，每层神经元的数量等来配置。我们已经在前面工作中使用的标准神经网络上进行了我们的实验来做对比。我们使用的网络是来自 [16] 的标准网络，我们称之为 FC100-100-10。这个神经网络有一个输入层，后面跟着两个隐藏层，每层都包含 100 个神经元，输出包含 10 个神经元的 softmax 层。

FC100-100-10 训练的学习率为 0.01，动量为 0.9，持续 500 个时期。每个小批量的大小为 500。在 MNIST 测试数据中，FC100-100-10 的分类准确率为 97.71‰

我们使用 Theano [41]，一个为多维数组的数学运算而优化的 Python 库，以及一个使用 Theano 的深度学习库 Lasagne [42]，来服务于我们在神经网络上的实验。

## 4.3 降维技术

主成分分析：我们使用 scikit-learn 的 PCA 模块 [40]。根据应用情况，可以指定要投影的组件数量或要保留的差异百分比。在向量化的 MNIST 训练数据上执行 PCA 以保留 99 % 的方差之后，缩减的维度为 331，这是我们针对 MNIST 数据集在基于 PCA 的防御的实验中使用的第一个减小的维度。对 HAR 训练数据执行 PCA 以保留 99 % 的方差后，缩小的维数为 155，这是我们针对 HAR 数据集在基于 PCA 的防御实验中使用的第一个减小的维度。

## 4.4 测量

### 4.4.1 对抗成功

对抗成功在针对导致错误分类的攻击的情况下，它的定义中存在微妙之处，而不是针对性的错误分类。在这种情况下，有 3 种可能的对抗成功概念可以被报告，我们在这里解释它。附录中解释了其他类似的概念。

对于每个良性输入 $\boldsymbol{x}$，我们检查两个条件：是否在干扰之后，$f(\widetilde{\boldsymbol{x}}) = f(\boldsymbol{x})$；是否最初，$f(\boldsymbol{x}) = y$，其中 $y$ 是真标签。因此，如果原始分类是正确的，但敌对样本的新分类是不正确的，则敌手的尝试是成功的。从某种意义上说，这个计数代表真正对抗的样本数量，因为它只是导致分类错误的对抗性扰动，并不是分类器在分类样本时的固有干扰。在统计敌对成功率的时候，我们将此计数除以通过整个分类管道后正确分类的良性样本总数。

在整个第 5 节中，当我们评估我们的防御表现时，我们将强调对手需要在特定预算的攻击成功率上达到指定的错误分类率的预算。这有几个原因。首先，这允许我们略微避开人类可探测的干扰量的问题。另外，正如第 5 节将会看到的，修改分类器输出所需的扰动级别通常相当集中于同一类别的示例。也就是说，在一些干扰预算下，对抗成功率迅速上升。这意味着成功率对预算的选择有些敏感，但所需预算的中位数相对稳定。

#### 4.4.2 干扰幅度

当解释“人类可以检测到的扰动”的概念时，区分两种变化很重要。首先这是一个扰动会导致人类错误分类的例子。这对任何分类器的性能提供了一个粗略的限制：对于一种敌对的修改，在一些参考标准上（在这种情况下，参考标准是一个人）产生一个不同类别的输入，这个输入将不会算作对手的成功，因为它的参考标准的标签已经改变。

虽然这是一个有用的限制，但这通常不是“可检测”的意思。另一个扰动水平是人类认识到不可能自然发生的事件。换句话说，即使原始类别可能更多或更少可识别，修改过的样本看起来已被篡改。我们在图 3 中使用来自经过各种攻击修改的 MNIST 数据集的图像来说明这一点。这些图像验证了我们进行实验时的干扰极限，因为我们证明了我们的防御对人类可检测范围内攻击的有效性。

当 ML 系统所有者选择或者在降维空间中操作时，可以通过将输入 ML 系统投影回原始的高维空间来检查对抗扰动的存在（对于图像数据）。图 3 中强调的一个关键点是，当采用防御措施时，对抗性扰动可以更容易地检测到。从简化子空间投影回原始像素空间的图像中的扰动清晰可见，这表明虽然 $\ell$ 的扰动距离相等，但它们在感知上距离更远。因此，在高维情况下在攻击开始被检测到的干扰水平下评估防御会导致保守估计其有效性。

## 5 实验结果

在本节中，我们将概述我们的实验结果。我们试图回答的主要问题有：

i) 我们的防御对策略性攻击是否有效?

ii) 我们的防御能够对抗 vanilla 攻击?

iii) 我们的防御能否作用在不同分类器上?

iv) 我们的防御是否推广到不同的数据集?

我们的评估结果证实了我们的防御在各种场景下的有效性，每种场景都有数据集、机器学习算法、攻击和降维算法的不同组合。对于每一组评估，我们改变分类管道的特定步骤并修复其他步骤。基本配置：我们首先考虑将 MNIST 数据集作为输入数据的分类流水线，将线性 SVM 作为我们的分类算法，将 PCA 作为我们的防御中使用的降维算法。由于我们将线性 SVM 作为分类器，因此我们评估其对使用 2.2 节中描述的线性 SVM 攻击生成的敌对样本的敏感性。下面我们对每个数据集来评估我们针对从测试集开始创建的对抗样本的防御。除非另有说明，否则所有防御结果均适用于完整的测试集。为了证明我们的防御不仅在这个基线和各种配置下是鲁棒的，所以我们系统地研究了它的影响，因为管道的每个组成部分以及攻击都被改变了。

（a）数字'9' 的良性和干扰图像（针对不具有防御的线性 SVM）：左边的第一张图像是原始图像，而其他图像是利用线性 SVM 的攻击（从左到右）修改的，$\xi = 0.5, 1.0, 1.5, 2.0$。干扰爬在 $\xi = 1.5$ 时开始可见，在 $\xi = 1.5$ 的图像中非常明显。攻击是在没有任何降维的分类器 $f$ 上进行的。

（b）数字'7' 的干扰图像（针对没有防御的神经网络）：图像通过对神经网络的快速梯度攻击（从左到右）进行修改，$\eta \approx 0.5, 1.0, 1.5, 2.0, 2.5$。在 $\eta = 1.5$ 时，再次开始可见，在 $\eta > 2.0$ 的图像中非常明显。攻击是在没有任何降维的分类器上进行的。

（c）数字'7' 的干扰图像（针对具有 $k = 70$ 的基于 PCA 的防御的神经网络）：图像已经通过神经网络上的快速梯度攻击进行修改，采用降维输入（从左到右），$\eta \approx 0.5, 1.0, 1.5, 2.0, 2.5$。将降维矢量投影回图像空间进行可视化。在这种情况下，干扰在 $\eta = 0.5$ 时开始可见，并且在 $\eta > 1.5$ 的图像中非常明显。这表明我们的防守也会使敌对扰动更容易被察觉。

图 3：为避开线性 SVM 和神经网络而生成的敌对图像

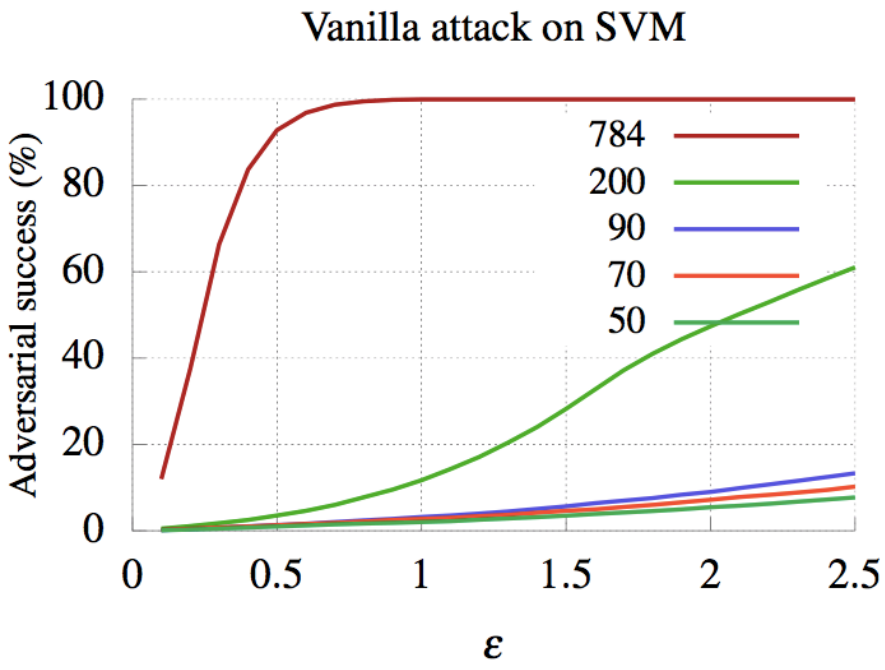


图 4：针对 MNIST 数据集的防御对线性 SVM 的 vanilla 攻击的有效性。MNIST 数据集上的敌对示例成功率与干扰程度 $\xi = ||\widetilde{\boldsymbol{x}} - \boldsymbol{x}||$ 的关系图。针对原始分类器进行攻击，并针对每个减小的维度 k 绘制防御的效果。

## 5.1 防御对支持向量机的影响

在标准情况下，我们首先回答问题 ii)，即“防御能否降低 vanilla 攻击的有效性?”和 i)，即针对线性 SVMs 的“防御能否降低策略攻击的有效性?”。

### 5.1.1 防御 vanilla 攻击

图 4 显示了成功防御对 SVMs 的防御攻击的变化。防御大大降低了敌对的成功率。例如，在 $\xi = 1.0$ 时，使用 $k = 50$ 的 PCA 的防御方法，使对手的成功率从 99.97% 降低到 1.85%。在 $\xi = 0.5$ 的情况下，敌对的成功率是 92.77% ,$k = 50$ 的防御将敌对成功率降低到 0.9% 这是两个数量级的下降。使用降维的数据训练会使得线性 SVMs 更健壮，这可以从防御的附加效果中看到。

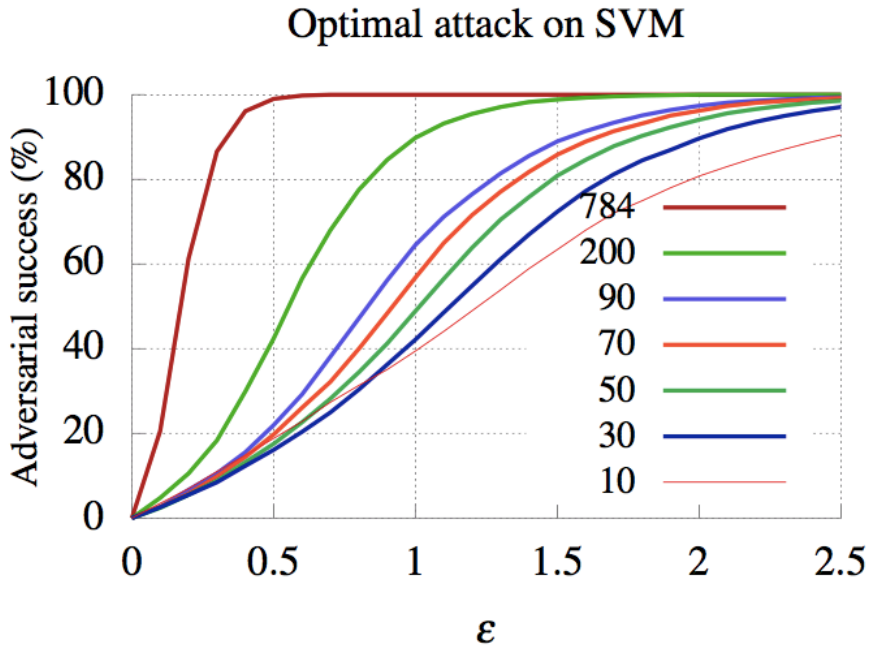


图 5：针对 MNIST 数据集的防御对线性 SVM 的最优攻击的有效性。将 MNIST 数据集上的干扰示例成功率与干扰幅度 $\xi = ||\widetilde{\boldsymbol{x}} - \boldsymbol{x}||$ 作图。针对每个降维分类器执行攻击并绘制防御效果。

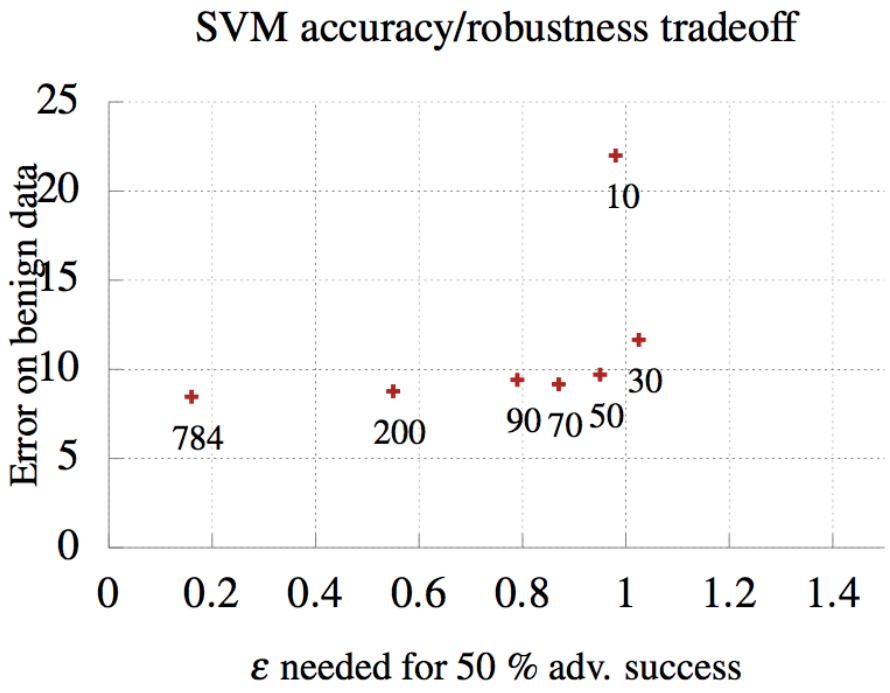


图 6：在良性测试数据和敌对性能之间的 SVM 分类性能之间的权衡。对抗性的健壮性是 $||\widetilde{\boldsymbol{x}} - \boldsymbol{x}||$ 的值，它允许对手达到 50% 的错误率。

同样，我们也注意到当我们减少在防御的投影步骤中使用的减少的维度 $k$ 时，对抗的成功率也在降低。在 $\xi = 1.0, k = 331$ 时，对抗成功率是 48.75%，当 $k = 100$ 时下降到 5.53%。在 $k = 30$ 时，对抗性的成功率下降到 2.63%，在 $k = 10$ 时减少到 2.52%。

在 vanilla 攻击下，防御就像一个噪音移除过程，消除了敌对的干扰并留下了干净的输入数据。与策略攻击相比，我们看到的是防御的鲁棒性。

### 5.1.2 防御对最佳攻击的影响

图 5 显示了针对线性 SVMs 的最优策略攻击的防御成功的变化。这个图对应的是对手意识到维度减少防御并将样本输入到管道中的情景，它的设计是为了最有效地避开降维分类器。在 0.5 的扰动程度下，没有防御的

分类器的分类错误率是 99.04%，$k = 70$ 的降维分类器的分类错误率只有 19.75%，即攻击成功率分别为 80.25% 和 5.01%。然而，由于 0.5 是一个小的干扰参数，即使当缩小的维度图像被投射回像素空间时，微扰也将是不可见的。在 1.3 的干扰参数中，开始清晰可见（见第 4.4 节），没有防御的分类器的错误分类率是 100%，大概是 77.11% 的分类错误率对于的 $k = 70$ 的降维分类器，几乎是降低了 23% 的攻击成功率。回想一下，避开低维度分类器所需要的扰动对人眼来说更清晰可见，使这些数字变得保守。

我们也可以研究我们的防御对达到一定的敌对成功率所需要的对抗预算的效果。为了达到 86.6%，需要一个 0.3 的预算，在没有防御的情况下进行分类，而对于一个 $k = 70$ 的分类器的所需预算是 1.6。对应的数字达到 90% 误分类率的 $k = 0.4$。因此，我们的防御很明显的降低一个非常强大的对手所进行的攻击的有效程度，它完全了解防御和分类器，并拥有最有效攻击的能力。

### 5.1.3 对防御的效用-安全权衡

图 6 显示了在普通和敌对条件下的性能之间的权衡。这个数据集的最佳维数显然在 50 到 30 之间，其中的扭结发生在这里。通过使用更多的维度，在分类性能方面几乎没有什么好处，而且使用更少的性能对健壮性没有任何好处。在 k=50 时，我们看到没有任何防御时测试集上的分类成功率下降了 91.5%, 而在防御下为 90.29%，因此，大约有 1.2% 的效用。

有了这些结果，我们就可以得出结论，我们的防御至少在基线情况下是有效的，对于线性 SVMs 的普通和最优攻击都是有效的。现在，为了证实我们关于我们的防御在机器学习分类器中的适用性的主张，我们研究了我们在神经网络上的防御表现。

## 5.2 防御对神经网络的影响

## 5.3 对不同数据集的适用性

接下来，我们通过更改所使用的数据集来修改基线配置。我们用线性 SVMs 作为分类器和 PCA 作为维度还原算法来显示结果。我们为人类活动识别数据集提供结果。

### 5.3.1 对 HAR 数据集的保护

在图 9 中，显示了由于防御而导致的攻击成功率降低。在 $\xi = 1.0$ 时，防御成功率从没有防御时的 99.56% 下降到 91.75%（$k = 70$）和 76.21%（$k = 30$）。为了达到分类错误率 90%，没有防御时需要的干扰程度是 0.65，在 $k = 70$ 时它增加到 $0 \cdot 876$，在 $k = 30$ 时为 1.26。因此，攻击方的成本增加了两倍以达到同样的成功率。这对效用的影响是适用的，在 $k = 70$ 时为下降了 2.3%，在 $k = 30$ 时为 5.4%，与安全获得的收益相比，这是微不足道的。

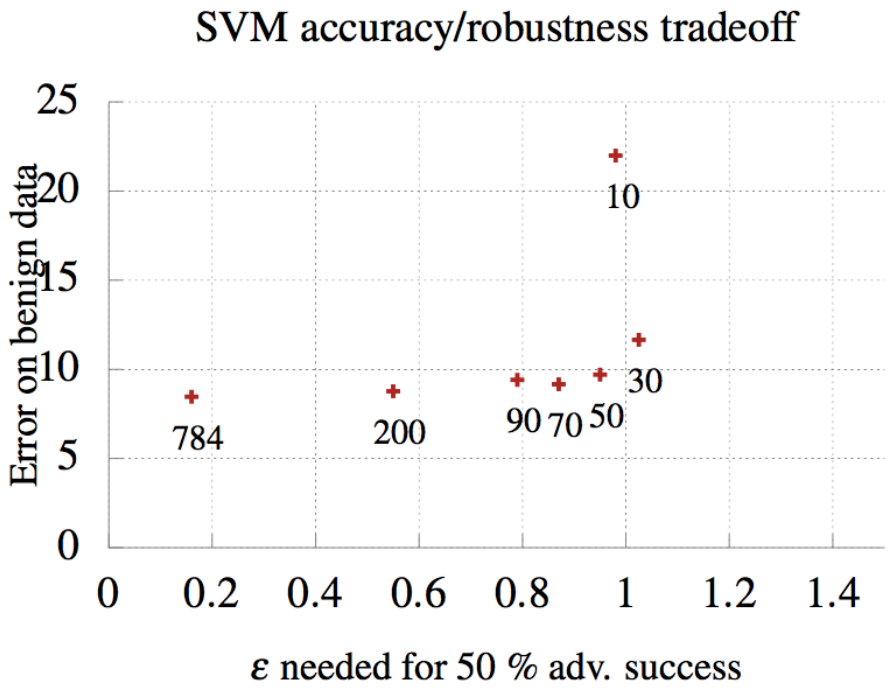


图 9：对于线性 SVM 攻击（针对原始分类器），HAR 数据集上的敌对示例与摄动幅度 e。针对防御中使用的每个减少的维度 k 绘制。

| | MNIST data | | | HAR data |
|---|---|---|---|---|
| | FC100-100-10 | Linear SVM | | Linear SVM |
| $k$ (MNIST) | | | $k$ (HAR) | |
| No D.R. | 97.47 | 91.52 | No D.R. | 96.67 |
| 784 | 97.32 | 91.54 | 561 | 96.57 |
| 331 | 97.35 | 91.37 | 200 | 96.61 |
| 200 | 97.04 | 91.28 | 100 | 92.43 |
| 100 | 97.36 | 90.89 | 90 | 94.60 |
| 90 | 97.14 | 90.58 | 80 | 94.54 |
| 80 | 97.25 | 90.64 | 70 | 94.37 |
| 70 | 97.52 | 90.76 | 60 | 93.72 |
| 60 | 97.38 | 90.47 | 50 | 92.47 |
| 50 | 97.26 | 90.18 | 40 | 92.06 |
| 40 | 96.71 | 89.03 | 30 | 91.11 |
| 30 | 96.56 | 88.37 | 20 | 88.63 |
| 20 | 96.67 | 86.69 | 10 | 86.67 |
| 10 | 93.22 | 77.79 | | |

表 2：降维防御的效用值。对于 MNIST 和 HAR 数据集，良性测试集的分类准确性针对用于基于 PCA 的防御的降维 k 的各种值以及没有防御的准确性提供。

## 5.4 对效用的影响

表 2 显示了我们的防御对良性数据的分类精度的影响。关键的结论是，神经网络和线性 SVMs 的精度降低到 k=50 的程度是最多 4%，此外，我们注意到，使用 PCA 的维数减少实际上可以提高分类精度，当 $k = 70$ 时，MNIST 数据集的准确性从 97.47% 降低到 97.52%，然而，更重要的维度减少，这将导致分类精度的急剧下降，这是意料之中的，因为用于分类的大部分信息都丢失，这些结果突出了我们在应用领域的防御的广泛适用性。很明显，我们防御的有效性并不是来自于 MNIST 数据集的特定结构的产物，他们对不同的数据都有影响。

# 6 讨论和限制

## 6.1 满足设计目标

在第 3.2 节中，我们列出了任何防御都应该具备的理想目标。首先，防御应该保持较高的分类精度。从表 2 中可以看出，对于数据集和分类器来说，有一系列缩小的维度对分类精度影响最小，在一些特定的情况下可以证明这一点。其次，基于 PCA 的防御版增加了样本数量 n 和维度 d 的多项式级别的代价。训练降维分类器所需的时间和空间高于高维空间中的分类器，因此，我们的防御在训练和测试阶段保持高效率。我们的防御所带来的附加安全已经在前一节的各种设置中得到了说明。从图 6 中可以清楚地看出，改变尺寸允许 ML 系统所有者在实用安全空间中导航不同的点。然而，当一个系统可能受到攻击时，我们的防御系统并不有效，这将导致我们在下面讨论的限制。

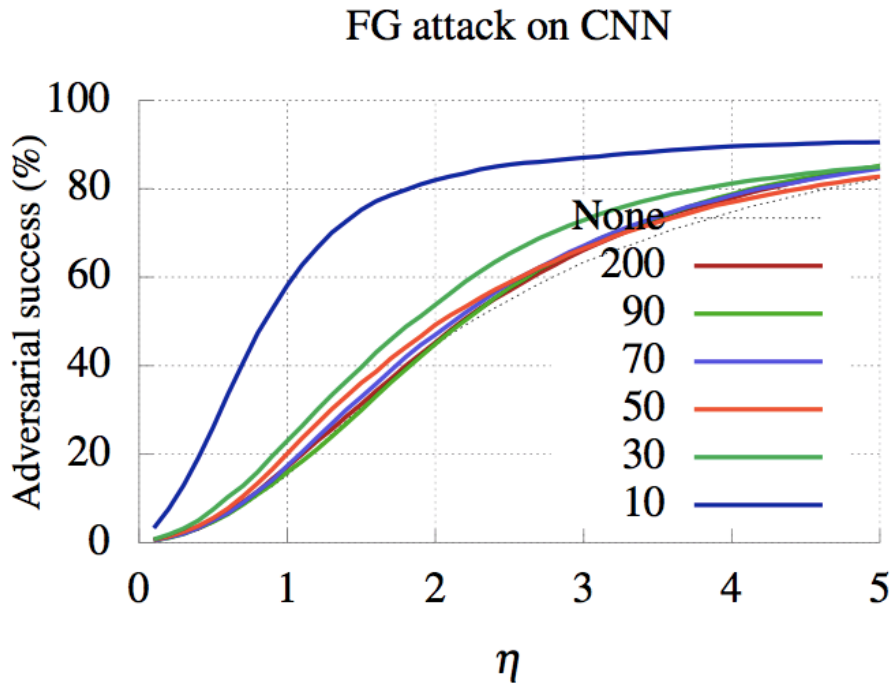


图 10：MNIST 数据集的防御效果与针对 Papernot-CNN 的战略性 FG 攻

击相抵触。在 MNIST 数据集上的对抗成功示例是相对于扰动幅度 $\eta = ||\widetilde{\boldsymbol{x}} - \boldsymbol{x}||$ 绘制的。针对每个子分类器（从算法 2 获得）进行攻击并绘制防御效果。

## 6.2 限制

尽管我们的防御在许多情况下降低了对抗性的成功率，但有两个主要的方面，它没有成为一种针对逃避攻击的全面防御机制：

1 在自己的不足：虽然我们的防守在各种情况下都能显著降低对手的成功率，但在某些情况下，对手的成功率仍然太高。在这种情况下，我们的防御系统很可能会被合并有其他的防御措施，如对抗训练［19］和整体方法［43］以建立一个针对逃避攻击的 ML 系统。我们的防御有一个优势，它可以与各种各样的 ML 分类器一起使用，它不会干扰其他防御机制的操作。此外，正如在第四部分示范的那样，我们的防御导致了一种具有更大视觉感知能力的“意即性”的混乱。这可能有助于防御的防御，目的是探测敌对的扰动。

2 缺乏普遍性: 在某些情况下，我们的防御能力有限。例如, 在图 10 中, 我们看到，基于 PCA 的防御系统几乎没有为 Papernot-CNN 提供安全改进（详情见第 9.3 条）。这一效应很可能源于这样一个事实，即 CNNs 已经在其卷积层中已经处于企业领域特定的知识，另外，使用 PCA 进行预处理的附加层不会带来任何额外的健壮性。此外，PCA 可能会减少 CNN 的卷积层用于分类目的的本地信息的数量。

解决我们防御的局限性的一个关键步骤是使用其他维度减少技术这可以将敌对的成功降低到可以忽略的水平，并与诸如 CNNs 这样的分类器结合在一起。在未来的工作中，我们计划探索减少维度的技术，例如自动编码器，内核 PCA 和各种压缩方案，以更好地理解维度减少与分类器的鲁棒性之间的关系。

# 7 相关工作

## 7.1 敌对机器学习

在垃圾邮件分类器的背景下，[44,45] 首先指出了机器学习算法对对抗性修改数据缺乏鲁棒性。[10-12] 对可能的攻击和敌对知识进行了明确的分类，其主要对抗目标是完整性，可用性和侵犯隐私。与当前工作最相关的目标是保证完整性，对手为了自己的利益试图导致数据错误分类。违规行为可以大致分为两类，即毒化攻击和规避攻击，本文集中讨论后者。在毒害攻击 [13,46-48] 中，攻击者在训练阶段之前或期间修改训练数据，以实现在测试时间规避等目标。

## 7.2 规避攻击

另一方面，在规避攻击中，攻击者的目标是构造 ML 系统错误分类的样本，她使用不同的关于系统的知识程度，并且只在测试时间进行分类。这些攻击已被提出用于各种机器学习分类器，如支持向量机 [14,17]，基于树的分类器 [17,18]，如随机森林和推动树，最近还有神经网络 [15,16,16]19-22]。使用机器学习的应用程序的脆弱性也得到了证明，例如人脸检测 [23]，语音命令识别 [24] 和 PDF 格式检测 [25] 等，这些都强调了防御的必要性。

## 7.3 分析回避攻击

有几个理论尝试来理解机器学习系统中存在漏洞的原因。纳尔逊等人 [33] 为了找到对抗样本，获得了分类查询数量对凸诱导分类器的界限。Fawzi 等人 [35，49] 研究随机噪声与对抗性扰动之间的关系，以便理解为什么分类器对随机噪声强健，但对对抗性扰动不敏感。

Tanay 等人 [50] 对于线性分类器，就数据子流形与分类器边界之间的距离而言，提供了敌对示例的几何透视图。

## 7.4 和过去的防御策略对比

以前有关对抗案例的防御工作主要集中在特定的分类器家族或应用领域。而且，现有的防御措施只是提高了安全性，仅仅抵御了文献中的现有攻击，而且不清楚防御机制是否能够有效地对抗知道其存在的对手，即利用防御弱点的战略攻击。作为一个例子，使用神经网络的精馏来对抗基于雅可比矩阵的显着图攻击 [15]。但是，Carlini 等人 [21] 表明，修改后的攻击使神经网络再次变得脆弱。现在，我们概述文献中现有的防御措施。

### 7.4.1 具体的分类器

Russu 等人 [26] 通过增加各种正则化来提出 SVM 的防御。Kantchelian 等人 [18] 提出了针对基于树的分类器而专门设计的最佳攻击的防御措施。现有的神经网络防御 [52-56] 进行了各种结构修改，以提高对示例的复原力。这些防御措施并不容易在整个分类器中普遍化，并且可能仍然容易受到副作用例子的影响，如 Gu 和 Rigazio [52] 所示。

### 7.4.2 具体的应用程序

Hendrycks 和 Gimpel [57] 研究将图像从 RGB 空间转换到 YUV 空间，以便更好地检测人体并降低错误分类率。他们还使用美白技术使 RGB 图像中的敌对扰动更易于人眼看到。还研究了 JPG 压缩对敌对图像的影响 [58]。他们的结论是，当扰动很小时，它有一个小的有益效果。这些方法仅限于打击针对图像数据的逃避攻击，并且根据其性质，不能在应用程序中进行通用化。

### 7.4.3 一般的防御

Smutz 和 Stavrou [43] 使用分类器集合来检测规避行为，通过检查不同分类器之间的分歧。然而，分类器的集合可能仍然容易受到广泛的例子的影响，因为它们在整个分类层次上进行了概括。此外，Goodfellow 等人 [19] 表明，集成方法对神经网络的逃避攻击的有效性有限。Goodfellow 等人 [19]

重新训练敌对样本以提高神经网络的韧性。他们发现这种方法在一定程度上减少了对手的成功，但仍然导致对对手样本的高置信度预测。在我们的实验中，我们发现重新训练敌对样本对提高线性支持向量机的鲁棒性影响极其有限，因此这种防御可能不适用于整个分类器。在 [59] 中，随机特征无效被用来减少逃避对神经网络攻击的敌对成功率。没有研究这种思想在分类器中的适用性。[60] 使用对抗性特征选择来提高 SVM 的鲁棒性。他们发现并保留了降低敌对成功率的功能。这种防御可能会在其他分类器中普遍化，并且是未来工作的有趣方向。

# 8 总结

在本文中，我们考虑了使用降维作为针对 ML 分类器的规避攻击的防御机制。我们的防御依赖于以下几点：（a）数据的维度降低可以作为降噪过程，有助于降低敌对扰动的幅度，（b）对降维数据进行训练分类器可以提高 ML 分类器的可靠性。

通过对多个实际数据集进行实证评估，我们证明了在一系列攻击策略（包括战略策略），ML 分类器和应用程序中，对抗成功率降低了 2 倍。我们的防御对分类器的效用（减少 1-2％）具有适度的影响，并且计算效率高。

因此，我们的工作为应对躲避攻击威胁提供了一个有力的基础。

# 致谢

我们要感谢 Chawin Sitawarin 在实验和讨论方面提供帮助。Arjun Nitin Bhagoji 由 NSF 和 DARPA 提供支持。Daniel Cullina 由 DARPA 支持。

# 参考

参考文献：略

# 9 附录

## 9.1 测量对抗成功

回想一下，我们在第 4 部分中使用了一种特殊的对抗性的成功。还有两个相关的概念可能被使用：

- 对于每一个 $\boldsymbol{x}$，我们检查 $y_a dv(= f(\boldsymbol{x}_a dv)) = f(\boldsymbol{x})$ 是否成立。这计算出了对抗样本的总数，其中的扰动会导致由分类器为干净的样本 $\boldsymbol{x}$ 分配的类别发生变化。可能会出现这样的情况，无论是干净的还是敌对的样本都没有被分配到正确的类别，因为分类器在测试装置上没有百分之百的精度（也可能在训练集上），然而，也可能是添加微扰导致分类器正确地对先前错误的输入进行分类。我们可能有 $f(\boldsymbol{x})not \in y$，但是 $y_a dv = y$，这是不太可能但可能发生的情况。

- 对于每一个 $\boldsymbol{x}$，我们检查是否 $y_a dv = y$。这计算了在摄动后的类不等于真正的类的敌对样本的总数。然而，这一数字还将包括那些具有对抗性的样本，而这些样本已经被错误地分类了。在这种情况下，不管攻击或防御的有效性如何，错误分类的对抗样本的百分比不能低于基线的分类器在良性测试样本上的不准确，这代表了对任何攻击的有效性的下限。

目前还不清楚这三种统计中哪一项在之前的工作中被认为是对方成功。我们在实验中计算了所有 3 项，发现它们是相似的。

## 9.2 用于防御评估的测试数据的直觉

使用的数据：规避攻击通常涉及到现有样本的修改化，使它们被错误地分类。如果一种攻击可以被认为是有效的，如果它导致了对来自训练集的敌对样本的高误分类率。由于各种原因，分类器在测试集上的准确性可能不高，而将敌对修改隔离为错误分类的原因可能是有问题的。此外，由于分类器通常是经过训练的，直到它们在训练集上有非常高的准确性，他们

的决策界限反映了培训数据的分布，一个涉及到最少修改训练数据的攻击是成功的。

我们对来自测试集的反式修改样本进行了评估，主要原因是在训练集上的过度拟合可能是防御效果的一个可能的原因，因此，对防御的准确评估应该包括从测试集中制作的对抗性样本。因此，一种防御机制使一个分类器更加安全，如果从测试集中的、经过修改的样本中，管道的分类精度高于原始分类器。精确度越高，防守就越精确。

## 9.3 CNNs

我们还在一个卷积神经网络 [38] 上进行实验我们从纸上获得的架构。这个 CNNs 的架构如下：它有两个卷积层，每个层有 32 个过滤器，后面是一个最大的池层，然后是另一个两个卷积层是 64 的过滤器, 然后是一个最大的池层。最后，我们有两个完全连接的层，每个层都有 200 个神经，然后是一个回归函数输出，有 10 个神经元（在 MNIST 的 10 个类中）隐藏层的所有神经元都是 ReLUs。我们把这个网络称为“网络报纸”。它的学习速度是 0.1（在过去的 10 年里调整到 0.01），并且 50 个时期的动量为 0.9。批大小是 MNIST 的 500 个样本，在 MNIST 测试数据上我们在 Papernot-CNN 网络得到了一个分类精度是 98.91%。