1. 、**应用实例：蓝盾AI防火墙**

按照模块化的思想，我们把基于AI的文档分类器作为一独立检测模组，集成到边界安全产品如防火墙中。一个拥有30多年历史的老式安全产品，该如何为其插上AI的翅膀？

在当今边界安全产品中（如防火墙），对于网络应用层（第七层）的恶意文件扫描已经是国际标准。业界对此功能的需求极为严苛，一个优秀的功能模组通常要求单个文件检测（扫描）时间（即延迟）维持在毫秒级别，对文件的检测准确率需在99%以上，而误报率则不能大于0.01%。

低延迟的功能要求是因为文件扫描功能需串联于整个文件检测的流水线中，高延迟会显著增加丢包率，造成严重数据丢失，这对于安全设备是不允许的。这些年，随着恶意文档的急速增加，先前基于规则匹配的引擎从标配开始变得力不从心：一方面为了达到产品需求的高准确度，需要大量的安全分析人员编写规则以更新之前的规则库；另一方面随着核心规则库的扩大，规则匹配算法的时间也呈现指数级别增长。这两方面都促使我们探索更新更好的AI引擎，并把此技术运用于实际工程中。

嵌入AI PDF分类器的智能防火墙

按照模块化的思想，我们把基于AI的PDF分类器看作一个独立的检测模组，并把此检测模块集成到安全产品防火墙中。一个面世30多年的老式安全产品，如何为其插上AI安全的翅膀，蓝盾在这方面走在业界的前列。

在当今边界安全产品中，如防火墙，对于网络应用层（第七层）的恶意文件扫描已经是国际标准。业界对此功能的需求极为严苛，一个优秀的功能模组通常要求单个文件检测（扫描）时间（即延迟）维持在毫秒级别，对PDF文件的检测准确度需在99%以上，而误报率则不能大于0.01%。

低延迟的功能要求是因为文件扫描功能，在硬件层，是串联于整个文件检测的流水线中，高延迟会显著增加丢包率，造成严重数据丢失，这对于安全设备是不可容忍的。这些年，随着恶意软件的急速增加，先前基于规则匹配的引擎从 标配 开始力不从心：一方面为了达到产品需求的高准确度，需要大量的安全分析人员编写规则和旧设备的规则库；另一方面随着核心规则库的扩大，规则匹配算法的时间也呈现指数级别增长。这两方面都促使我们探索更新更好的AI引擎。

综上所述，我们把基于AI的PDF分类器串联集成到防火墙中，成功替代了原本基于规则的引擎。纵然上述两引擎均基于静态分析技术，AI引擎带来的优势却是非常明显的。一方面，此引擎本无需频繁更新，据实验数据，正常AI引擎的平均更新周期为半年，而规则引擎则为2周；第二方面是AI引擎卓越的低计算资源消耗，据实验数据，AI引擎能在预测时稳定于约1/3的CPU占用，约50%的内存占用。CPU计算力的消耗主要在特征提取和计算最终结果概率上，内存的消耗则来自AI模型自身在预测时需位于内存中。

在此真实防火墙应用场景中，我们根据AI模型不同的输出概率以及原因，分情况触发后续多样的动作。举例说，当输出概率大于0.9时，表明基于静态分析方法有较大把握预测此样本为正（即恶意），故会第一时间触发安全产品，如防火墙，对此所属链接进行封锁（block）操作；相反地，当概率小于0.1时，表明模型有较大把握预测此样本为负（即善意），安全产品默认允许此所属链接。

有趣的是当输出概率在灰色区间时如[0.1, 0.9]所触发的行为，我们会把样本上传到云中作动态分析，此分析包括沙箱，安全分析师和威胁情报等。静态分析的优势是速度快，但对于某些具有强逃逸能力（如混淆加密等）的样本会有一定的局限；动态分析的优势是准确度高，分析深入但速度慢，占用资源较高。

综上所述，我们可知，基于AI的静态分析和动态分析技术各有千秋，我们通过把两种技术合理运用，能较好地满足真实应用场景下的产品需求。