基于 Kitsune 入侵检测系统的可解释器

刘旺

摘要

由于能够检测不可预见的威胁的巨大前景以及深度神经网络（DNN）提供的卓越性能，无监督深度学习（DL）技术已广泛应用于各种与安全相关的异常检测应用中。然而，缺乏可解释性给深度学习模型在实践中的采用造成了关键障碍。现有的解释方法是针对监督学习模型或者非安全领域提出的，它们不适合无监督深度学习模型，并且无法满足安全领域的特殊要求。在本文中，作者提出了 DeepAID，一个通用框架，旨在（1）解释安全领域中基于深度学习的异常检测系统，以及（2）基于解释提高这些系统的实用性。作者首先通过制定和解决具有安全域特殊约束的精心设计的优化问题，提出了一种新的无监督 DNN 解释方法。然后，作者提供了几个基于 Interpreter 的应用程序，以通过解决特定领域的问题来改进安全系统。作者将 DeepAID 应用于三种类型的安全相关异常检测系统，并通过具有代表性的先前工作广泛评估提出的解释器。实验结果表明，DeepAID 可以为无监督深度学习模型提供高质量的解释，同时满足安全领域的一些特殊要求。本文还提供了几个用例来表明 DeepAID 可以帮助安全操作员理解模型决策、诊断系统错误、向模型提供反馈并减少误报。

关键词：深度学习；入侵检测；模型可解释性；无监督学习

1 引言

到目前为止，无监督深度学习模型已应用于各种与安全相关的异常检测应用，例如网络入侵检测、系统日志异常检测、高级持续威胁（APT）检测、域生成算法（DGA）检测和网络攻击检测。尽管展示了巨大的前景和卓越的性能，但深度学习模型，特别是深度神经网络 (DNN)，其决策缺乏透明度和可解释性。黑盒问题给深度学习模型在实践中造成了阻碍，尤其是在安全相关领域。首先，如果没有充分的理由和可信的证据，很难从简单的二元（异常或正常）结果建立对系统决策的信任。其次，基于深度学习的黑盒系统很难与专业知识相结合，难以排除和调试决策错误或系统错误。第三，减少误报（FP）是异常检测系统在实践中最具挑战性的问题。如果不了解模型的工作原理，就不可能更新和调整 DL 模型以减少 FP。

2 相关工作

近年来，一些研究试图通过找出对最终决策影响最大的一小部分特征来开发解释深度学习模型决策的技术。然而，由于两个原因，这些技术不能直接应用基于深度学习的异常检测。首先，诸如[1][2], 之类的现有研究主要集中在解释深度学习模型以进行监督分类，而不是无监督异常检测。由于这两类学习方法的机制根本不同，直接应用这类技术是不合适且无效的。其次，大多数现有方法都是为计算机视觉等非安全领域设计的[3]，以理解 DNN 的机制。但是，安全从业者更关心如何根据解释设计出更可靠、更实用的系统。此外，研究表明，现有方法由于性能较差而未能在安全领域采用[4]。

2.1 基于近似的解释

这些方法使用相当简单且可解释的模型来局部近似原始 DNN 的某些决策。这项技术基于这样的假设：尽管 DNN 中的映射函数极其复杂，但特定样本的决策边界可以简单地近似（例如，通过线性模

型来近似）。然后，通过简单模型提供样本的解释。例如，LIME[5]使用解释样本的一组邻居来训练可解释的线性回归模型以拟合原始 DNN 的局部边界。LEMNA 是使用在安全领域循环神经网络（RNN）的方法，与 LIME 不同，LEMNA 中使用的替代可解释模型是非线性混合回归模型。基于近似的解释方法在实践中具有一定的局限性。首先，它们只提供了对模型行为的局部近似解释，无法全面解释模型的整体行为。其次，由于简化了模型结构，这些方法可能忽略了深度学习模型中复杂的非线性关系和交互效应。

2.2 基于扰动的解释

基于扰动的解释是一种常见的解释深度学习模型的方法，它通过对输入样本进行微小的扰动来分析模型的响应和预测结果的变化。这种方法可以帮助我们理解模型对输入的敏感性和鲁棒性。

基于扰动的解释方法通常包括以下几个步骤：

扰动生成：选择一个输入样本，并在其附近引入一些微小的扰动。这些扰动可以是随机的，也可以是根据某种规则生成的。

扰动应用：将扰动应用于输入样本，得到扰动后的样本。扰动可以通过添加噪声、修改像素值或应用其他变换方式来实现。

模型响应分析：使用扰动后的样本作为输入，观察模型的预测结果和响应的变化。可以分析预测的置信度、类别分数或激活响应等。

解释生成：根据模型响应的变化，生成对模型行为的解释或解释性指标。这可以包括识别模型对哪些特征或区域特别敏感，或者分析模型对不同类别的决策边界。

基于扰动的解释方法可以提供关于模型行为的有用见解，例如，模型对输入样本的敏感程度、对噪声的抗干扰能力、对不同特征的关注程度等。此外，通过分析不同类型的扰动，我们还可以了解模型的鲁棒性和可解释性的稳定性。

一种常见的基于扰动的解释方法是输入空间扰动（Input Space Perturbation）。该方法通过在输入样本中添加噪声或修改像素值来生成扰动样本，然后通过观察模型对扰动样本的响应来推断模型对输入的敏感性和预测结果的变化。

除了输入空间扰动，还有其他类型的扰动方法，如特征空间扰动（Feature Space Perturbation）和激活空间扰动（Activation Space Perturbation）。这些方法根据不同的应用场景和需求，以不同的方式引入扰动并分析模型的响应。

2.3 基于反向传播的解释

基于反向传播的解释方法是一种常见的解释深度学习模型的方法，它利用模型的反向传播过程来分析模型的预测结果和对输入的敏感性。该方法可以帮助我们理解模型的决策过程以及不同输入特征对于预测结果的贡献程度。

基于反向传播的解释方法通常包括以下几个步骤：

前向传播：将输入样本通过深度学习模型进行前向传播，得到预测结果。在这个过程中，每一层的输出和激活函数都被记录下来。

反向传播：根据预测结果，通过反向传播算法计算模型中每个参数的梯度。这个过程会从最后一层开始，逐层向前计算梯度，并记录每个梯度值。

梯度分析：根据反向传播过程中计算得到的梯度，可以分析不同输入特征对预测结果的影响程度。一种常见的方法是使用梯度的绝对值或平方作为特征的重要性指标，较大的梯度表示该特征对预测结果的贡献较大。

可视化和解释：通过可视化梯度或梯度相关的信息，可以将模型的解释结果呈现出来。例如，可以将梯度映射到输入图像的像素空间，从而可视化模型对不同区域的关注程度。

基于反向传播的解释方法可以提供关于模型预测结果和输入特征之间的关系的一些直观理解。通过分析梯度，我们可以了解模型对输入的敏感性，以及哪些特征对于模型的决策具有较大的影响。

需要注意的是，基于反向传播的解释方法也有一些局限性。模型的梯度信息可能受到一些限制，例如梯度消失或梯度爆炸等问题。此外，梯度信息可能无法完全捕捉到模型的复杂决策过程，特别是在深层网络中。

3 本文方法

在本文中，作者开发了 DeepAID[6]，这是一个解释和改进安全应用中基于深度学习的异常检测的通用框架。DeepAID 的高层设计目标包括（1）为无监督深度学习模型开发一种新颖的解释方法，满足安全领域的一些特殊要求（如高保真、人类可读、稳定、鲁棒和高速），（2）解决安全系统的一些特定领域的问题（例如决策理解、模型诊断和调整、减少 FP）。

3.1 本文方法概述

此部分对本文将要复现的工作进行概述，图的插入如图 1所示：本文的 Interpreter 提供对无监督深度学习模型中某些异常的解释，以帮助安全从业人员理解异常发生的原因。具体来说，我们将对异常的解释表述为解决搜索正常“参考”的优化问题，并求出其与异常之间最有效的差异。提出了几种特定和解决优化问题的技术，以确保我们的解释能够满足安全领域的特殊要求。本文主要复现的工作有两部分：第一部分是基于无监督深度学习的入侵检测系统；第二部分是前一部分入侵检测系统的解释器。系统的工作流程图如图 1所示：

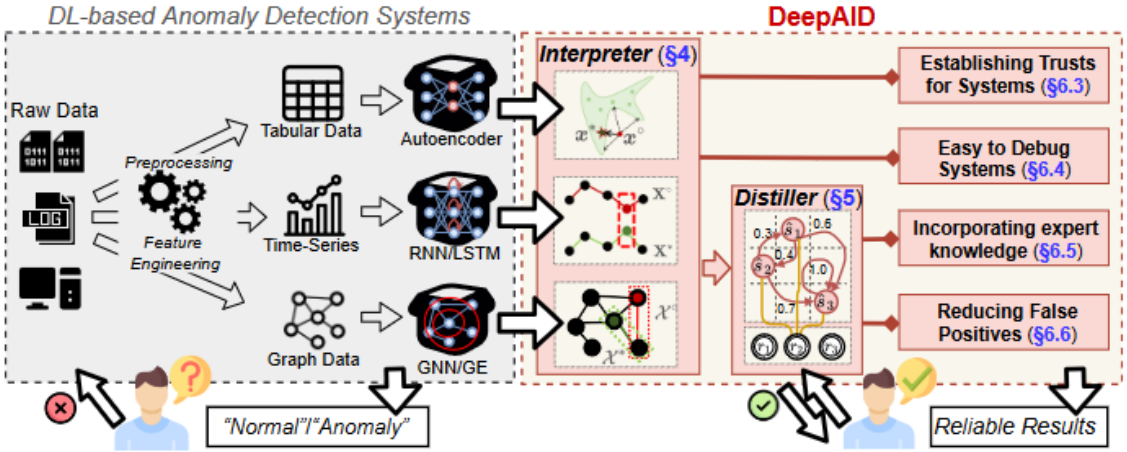

图 1: 工作流程图

3.2 入侵检测系统模块

本文复现的目标入侵检测系统主要是基于无监督学习方式，再进行细分就是属于基于重构的学习方式，代表的深度学习模型主要有 AutoEncoder 和 GAN 等；用于异常检测的无监督深度学习是用纯正常数据进行训练的，因此也称为“zero-positive”学习。现有的方法使 DNN 能够学习正常数据的分

布并发现偏离分布的异常，检测的数据类型是表格型，也就是一个数据的实例是一维的向量。本文所使用入侵检测系统 Kisune 如图 2所示：

3.3 入侵检测系统解释器模块

根据输入入侵检测系统的数据类型不同，可以将入侵检测系统分为三类：1）基于表格数据的系统；2）基于时间序列数据的系统；3）基于图数据的系统。根据入侵检测系统的不同，相应解释器模块也要进行更改，由于上文提到，复现的入侵检测系统是基于表格数据的，因此此处的解释器模块也选择相应的版本进行复现。

4 复现细节

本次复现工作主要引用了 https://github.com/dongtsi/DeepAID 中的代码。

4.1 入侵检测系统模块实现

Kitsune[7]系统中，特征提取和映射是通过多个步骤实现的，包括特征提取器（FE）、特征映射器（FM）和异常检测器（AD）三个组件。

特征提取器（FE）：FE 旨在实现对动态数据流或网络通道中的时序统计信息的高速提取。它捕获网络中每个数据包的上下文和目的，识别不同通道的数据包是交错的，并且在任何给定时刻都可能有许多通道。该框架使用在阻尼窗口上维护的增量统计，这意味着提取的特征是时序的，能够捕获近期行为。这种方法具有小的内存占用和 O(1) 的复杂度，因为增量统计的收集是通过哈希表维护的。它还维护了 2D 统计数据，捕获连接的接收和发送流量之间的关系。

特征映射器（FM）：FM 的作用是将 FE 中的 n 个特征映射成 k 个较小的子实例，每个子实例对应 AD 的 Ensemble Layer 中的一个自编码器。这种映射确保每个子实例 v 最多只有 m 个特征（其中 m 是用户定义的参数），并且 x 中的每个 n 个特征都恰好映射到 v 中的特征一次。这种映射被设计为足以捕获正常行为，以便检测相应子空间中的异常事件。这种映射是通过对 X 的特征进行递增聚类，将其分成 k 组，每组不大于 m，使用在递增更新的摘要数据上的聚类层次聚类来找到。

异常检测器（AD）：AD 包含一个特殊的神经网络，称为 KitNET，它是为在线异常检测而设计的无监督 ANN。KitNET 由两层自编码器组成：Ensemble Layer 和 Output Layer。Ensemble Layer 由一组 k 个三层自编码器组成，这些自编码器映射到 v 中的相应实例。这些自编码器测量 v 中每个子空间的独立异常性，并在训练模式期间学习其各自子空间的正常行为。在训练模式和执行模式期间，每个自编码器都会向 Output Layer 报告其 RMSE 重构误差。Output Layer 也是一个三层自编码器，学习 Ensemble Layer 的正常 RMSE，并通过考虑子空间异常之间的关系和网络流量中自然发生的噪声来产生最终的异常分数。

4.2 入侵检测系统解释器模块实现

在 DeepAID 中，解释器的实现主要集中在解释异常行为。这篇文章提出了一种统一的问题表述，用于解释无监督深度学习模型中的异常。重点在于探究为什么异常在深度学习模型中偏离正常数据。解释器采用“与参考值的偏差”来解释异常的偏离。具体来说，对于一个异常 x^0, 解释的目标是找到一个被深度学习模型认为是正常的参考值 x^*, 然后通过指出这个异常与其参考值之间的差异来得到对该异常的解释。对于安全领域的特殊要求，解释器需要满足以下几点：1. 保真度：需要高度保真地模

仿原始的深度神经网络（DNN）；2. 简洁性：结果应该是简洁且易于人类理解的；3. 稳定性：对于同一样本的结果应该是一致的；4. 鲁棒性：应对抗性攻击或噪声具有鲁棒性；5. 效率：解释应该在不延迟高速在线系统工作流的情况下立即可用。为了满足以上要求，文章提出了一个优化问题来形式化地表述对 x^0 的解释。这个问题的目标是最小化 x^* 的损失函数，该函数测量保真度、稳定性和简洁性的损失，并通过权重系数 1 和 2 进行平衡。此外，存在一个约束条件，即 x^0 必须在与异常 x^* 相同的特征空间中搜索。优化问题的具体函数公式可以用如下公式来表示。

$$arg\min_{x^*} ReLU\left(\varepsilon_R\left(x^*, f_R\left(x^*\right)\right) - \left(t^R - \varepsilon\right)\right) + \lambda \left\|x^* - x^o\right\|_2$$

公式中 $ReLU()$ 指的是 Rectified Linear Unit，ε_R 指的是保真度损失，$f_R()$ 指的是入侵检测模型，t_R 指的是模型的决策边界，ε 给定的很小的数，加号后面的一项是稳定性损失。

4.3 实验环境搭建

本次实验所用的实验平台的硬件情况是：处理器 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz，内存 48G，显卡为 GTX1080Ti，使用的操作系统为 Ubuntu 20.04.4 LTS，python3.9.0，pytorch1.7.1. 为了减小计算资源的使用，在特征映射时，我限定每个组的特征不会多于 10 个。

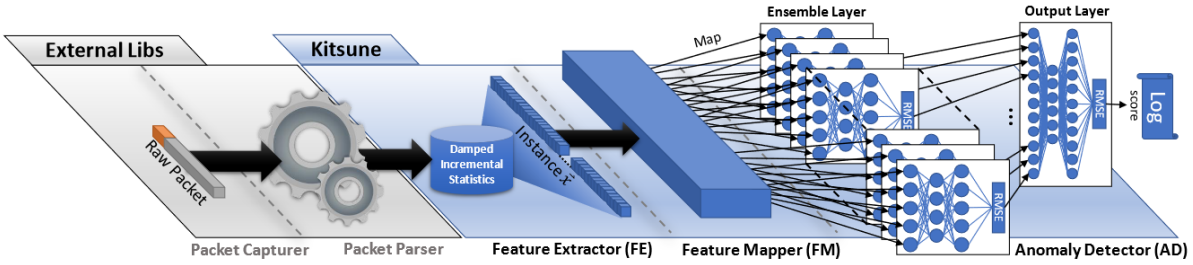

图 2: Kitsune 结构图

4.4 创新点

原论文中，作者所使用的入侵检测系统仅仅是一个很简单的 Auto Encoder，使用提出的解释器解释这个 Auto Encoder 所检测到的异常。在本文中，我使用 Kitsune 入侵检测系统替换了原有的简单的入侵检测系统，可以进一步检测解释器的泛化性和通用性。

5 实验结果分析

本次实验的数据集分为训练集和测试集，训练集和测试集里面各有 50000 条实例。训练集中的 50000 条实例全都是正常流量中采集而来的实例，测试集中的实例全是异常流量中采集而来的，也就是说使用了 zero-positive learning 的方式进行训练。

5.1 特征映射器的训练

在训练特征映射器的时候，我使用训练集中前 4096 条实例来映射器对 100 维的向量进行分组，每个组内的特征不超过 10 个。特征映射器的结构图如 3所示，列表中前两个 10 的含义为第一组和第二组中所包含的特征个数为 10 个。

```
the format of featuremapper is [10, 10, 5, 5, 10, 9, 7, 10, 6, 10, 5, 4, 4, 5]
```

图 3: 特征映射器的结构图

5.2 Kitsune 入侵检测系统的训练

在训练完映射器之后，下一步就是要训练 KitNET。KitNET 分为 ensemble layer 和 head layer，前者是多个三层的 Auto Encoder 组成的网络，具体有多少个要根据映射器的结构来决定，后者是一个三层的 Auto Encoder，用来对重建的结果打分。我使用训练集中剩下的所有实例（排除了用来训练映射器的 4096 个实例）来训练 KitNET。图 4是训练 KitNET 时，head layer 的损失变化，可以明显的观察到，head layer 随着训练时间的增加在逐渐收敛。

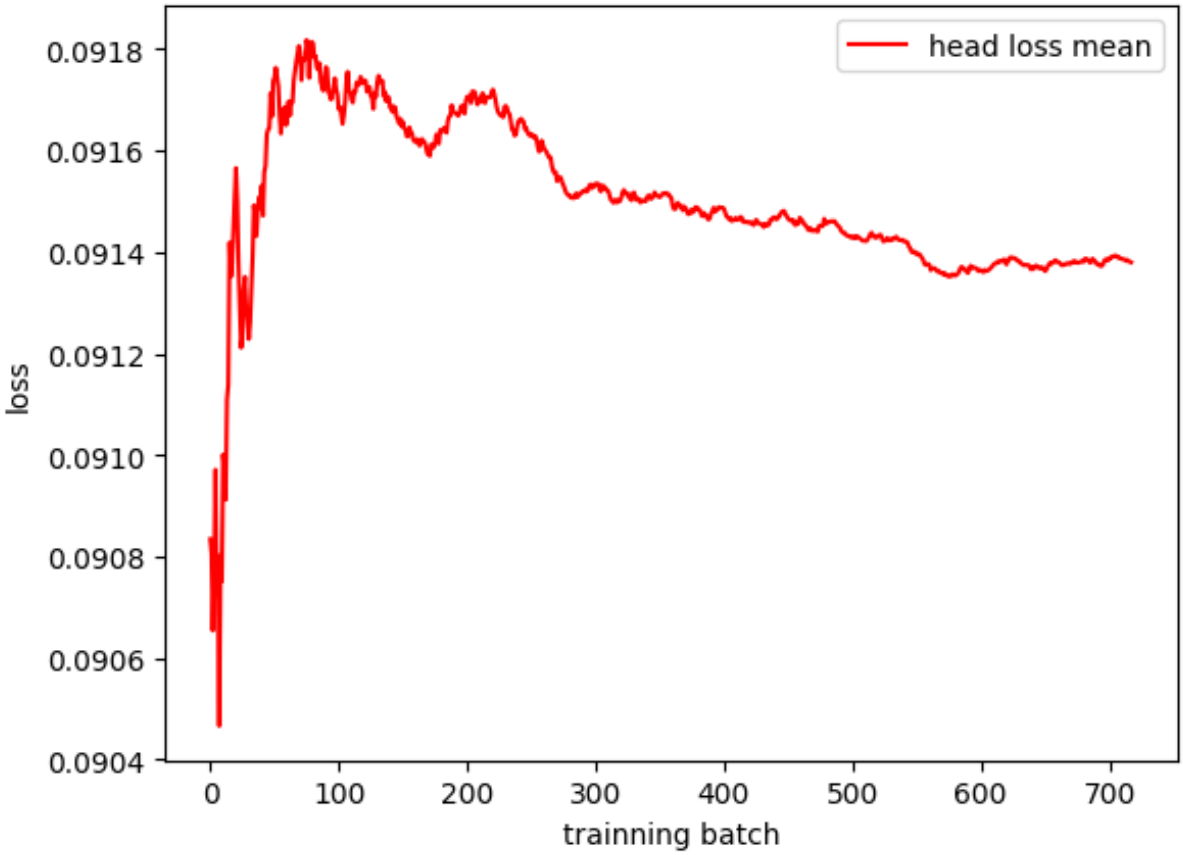

图 4: 特征映射器的结构图

训练完 KitNET 后，我们再来测试一下它的预测精度。我使用了测试集中所有的实例来进行测试，测试结果如图 5所示，图中黑色的线为入侵检测模型的决策边界，在黑线上方的红点是被模型判断为异常的实例，下方的红点被模型判断为正常实例，经过统计模型的精度有 0.9887。

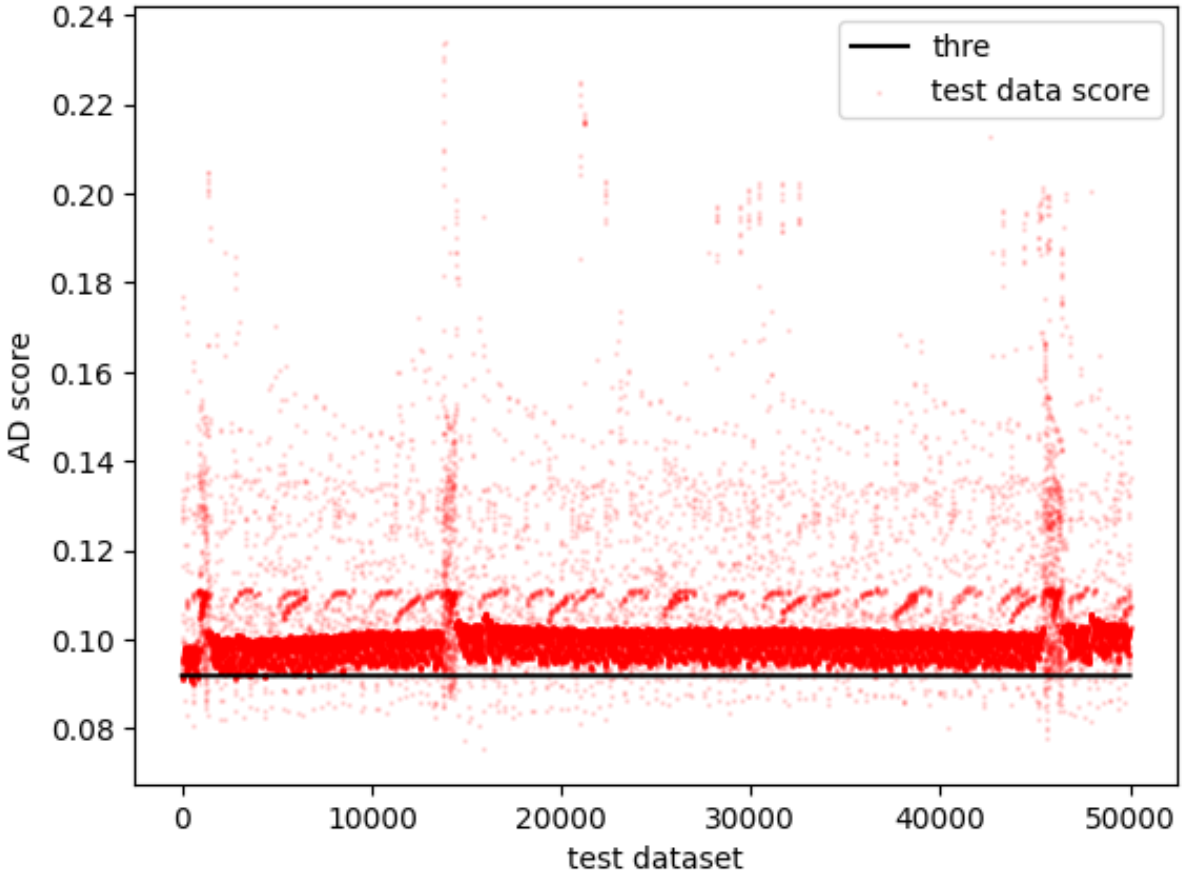

图 5: 特征映射器的结构图

5.3 解释器的训练

接下来对解释器进行训练。在对解释器进行训练时，先要将训练好的 KitNET 和它的决策边界输入解释器，然后用训练集中的异常数据进行训练。图 6是解释器损失的在训练时候的变化。可以清楚的观察到，解释器很明显的在逐渐收敛。

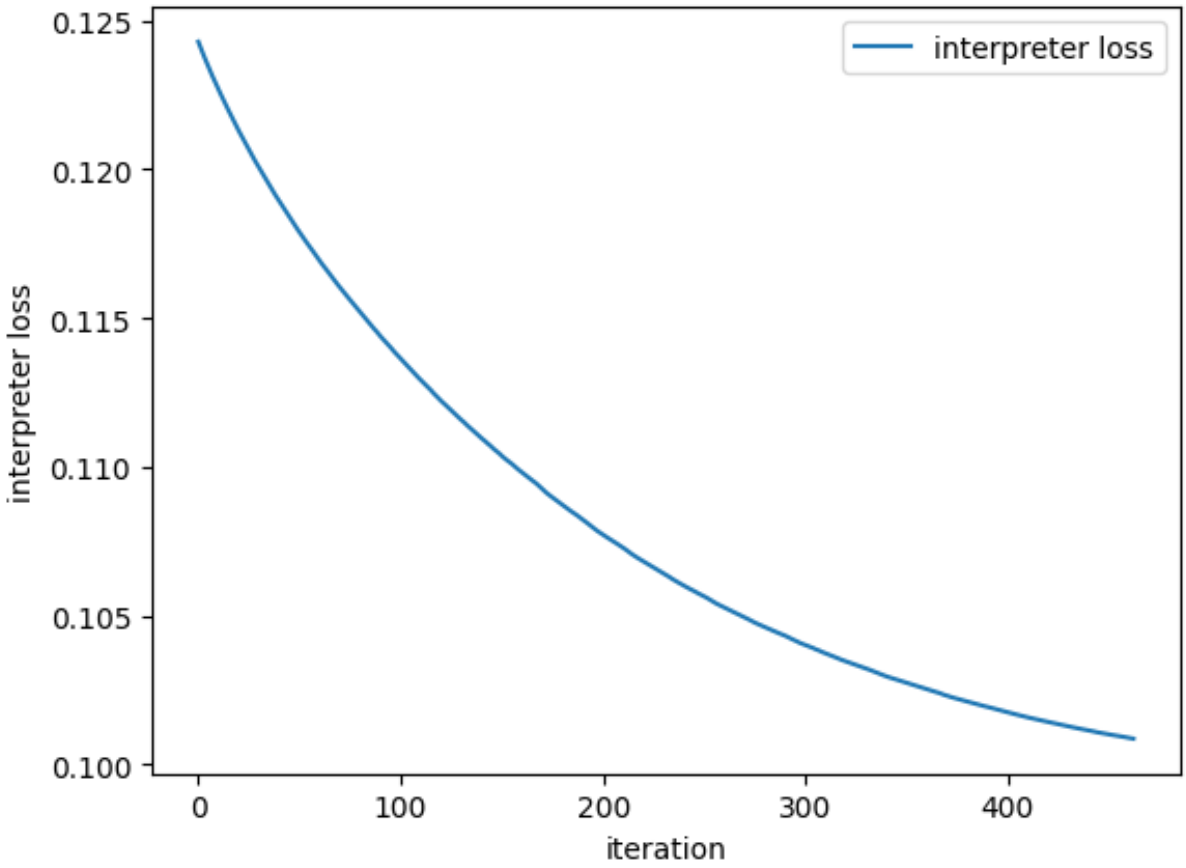

图 6: 特征映射器的结构图

解释器训练好后，再来看一看解释器的表现。在对异常进行解释的时候，解释器会选取 5 个差异最大的特征进行展示和比较，解释结果如图 7所示。可以看到，解释器所产生的参照和异常本身对比，差异还是十分明显。因为数值大小的原因，前两个特征的参照和异常的对比可能不明显，图 8是这五个特征中将参照比上异常本身值的比例图。

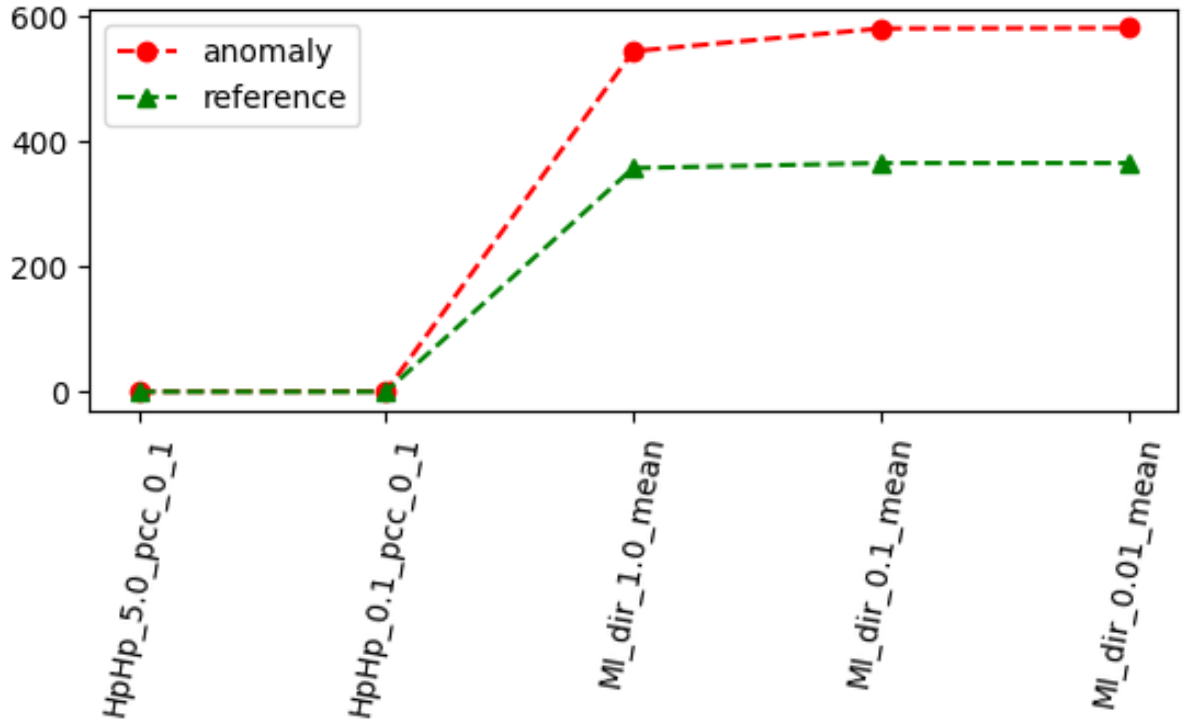

图 7: 特征映射器的结构图

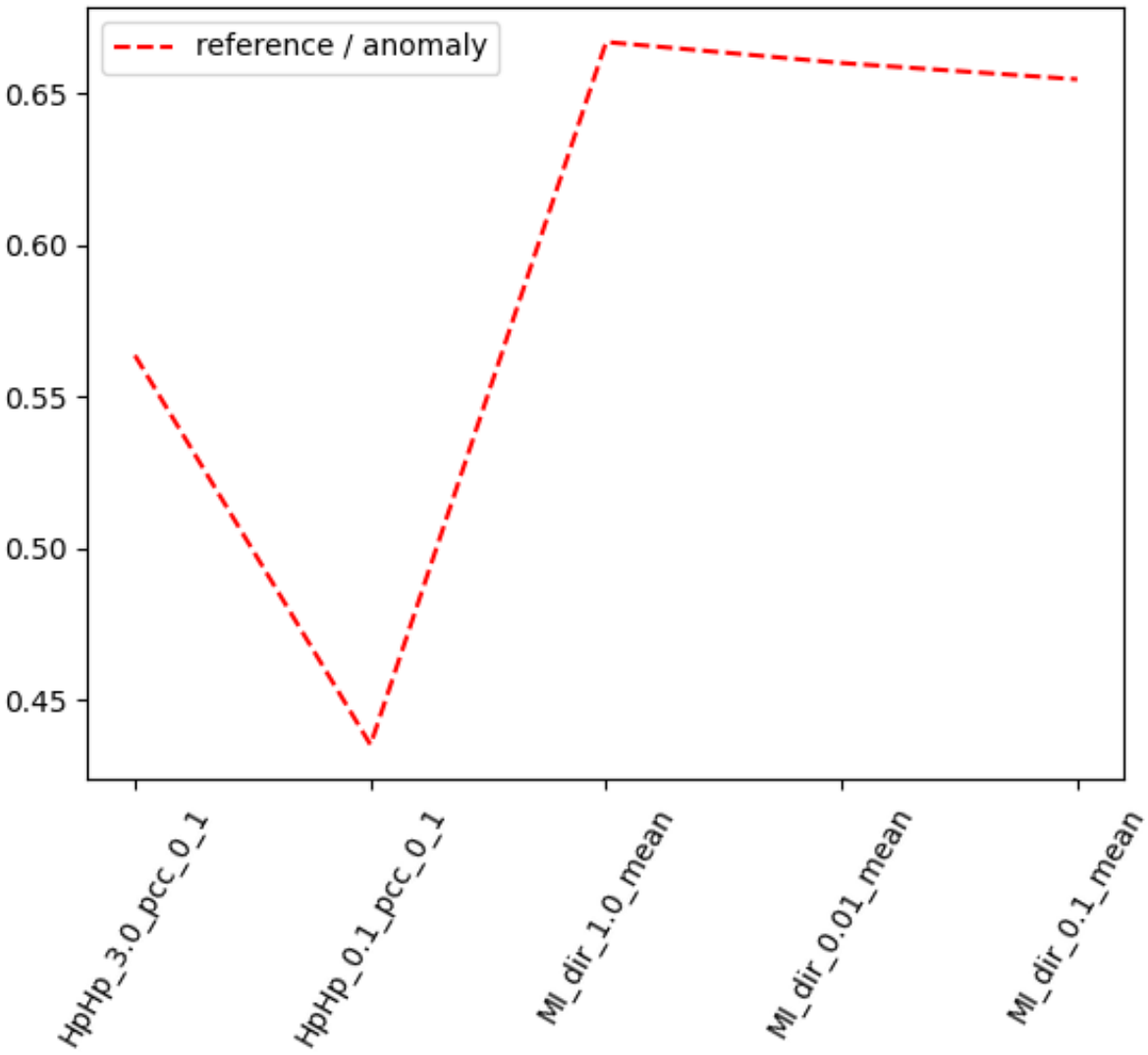

图 8: 特征映射器的结构图

6 总结与展望

本文首先对现有的深度学习模型可解释研究做了简要的分类和介绍，说明了不同类别可解释技术的原理。然后说明这些可解释技术在网络安全领域和无监督学习领域的不足，进而根据在安全领域中几个重要的指标，作者提出了一个解释器，用于增强模型的可解释性。在复现过程中有几个不足的地方。第一，对于入侵检测系统的实现，我只复现了处理表格数据的系统，还有处理时间序列和图数据的系统没有实现，因此相应的解释器也只实现针对了处理表格数据的系统。第二，关于实验数据部分，因为所选论文没有使用公共数据集，而是在一个小型网络中使用软件采集的网络数据，然后使用采集的数据作为数据集使用。在复现过程中，本来也应该先在相同的网络环境中采集数据，然后再进行实验的。可是由于时间和设备的原因，在复现过程中我未能自己采集数据，而是直接使用论文所提供的数据进行实验；并且由于实验数据不足，所进行的实验也不够多样化。

参考文献

[1] AMARASINGHE K, KENNEY K, MANIC M. Toward Explainable Deep Neural Network Based Anomaly Detection[C]/ /2018 11th International Conference on Human System Interaction (HSI). 2018: 311-317. DOI: 10.1109/HSI.2018.8430788.

[2] ANTWARG L, SHAPIRA B, ROKACH L. Explaining Anomalies Detected by Autoencoders Using SHAP[J/OL]. ArXiv, 2019, abs/1903.02407. https://api.semanticscholar.org/CorpusID:70349910.

[3] FONG R C, VEDALDI A. Interpretable Explanations of Black Boxes by Meaningful Perturbation[C]//Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2017.

[4] FAN M, WEI W, XIE X, et al. Can We Trust Your Explanations? Sanity Checks for Interpreters in Android Malware Analysis[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 838-853. DOI: 10.1109/TIFS.2020.3021924.

[5] RIBEIRO M T, SINGH S, GUESTRIN C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier[C/OL]//KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA: Association for Computing Machinery, 2016: 1135-1144. https://doi.org/10.1145/2939672.2939778. DOI: 10.1145/2939672.2939778.

[6] HAN D, WANG Z, CHEN W, et al. DeepAID: Interpreting and Improving Deep Learning-Based Anomaly Detection in Security Applications[C/OL]//CCS '21: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual Event, Republic of Korea: Association for Computing Machinery, 2021: 3197-3217. https://doi.org/10.1145/3460120.3484589. DOI: 10.1145/3460120.3484589.

[7] MIRSKY Y, DOITSHMAN T, ELOVICI Y, et al. Kitsune: an ensemble of autoencoders for online network intrusion detection[J]. arXiv preprint arXiv:1802.09089, 2018.