**基于深度神经演化的车辆CAN报文入侵检测算法**

随着车辆电气化和网络化的普及，车辆上搭载了越来越多的电子设备，与此同时必须要确保车辆各模块之间通信的安全性和可靠性来保证车辆驾驶员的人身安全。控制器局域网（CAN）通信是车辆上最常用的通信协议，该协议本身没有安全校验机制，这对于车辆上的通信安全构成了极大的威胁。神经结构搜索（NAS）是一种自动化设计网络结构的方法，可以以最低的搜索成本找到最优秀的神经网络架构。本文提出了一种基于深度神经演化的入侵检测系统，通过演化算法对神经网络的超参数进行优化以保证入侵检测系统的准确性的前提下最小化神经网络的复杂度。算法分为强化学习网络构成的CAN数据包获取部分和图网络与卷积网络构成的CAN报文检测部分。强化学习可以根据图网络中图卷积操作后的特征向量来学习车辆状态以便动态的获取每一次检测CAN样本的数据包。卷积网络和图网络分别用来提取CAN数据包的逻辑特征和空间特征来进行检测。本工作还提出一种利用车辆CAN报文的时序特征转换为图数据的一种方法，并且使用公开的数据集对提出的算法进行了验证。实验结果表明，与其他的深度学习算法相比，本算法在保证尽可能低的复杂度的同时达到了99%以上的检测准确率。

**关键词：**CAN报文入侵检测、多目标优化、深度神经演化、图神经网络、卷积神经网络

**摘要**

在过去几十年中，汽车电子产品的应用快速增长[[1](#_ENREF_1)] [1]，而这种趋势导致车辆生态系统发生了一些变化。例如，线控驱动（DBW）技术在控制系统中使用电子或电气系统，如油门、制动器和转向系统，这些系统传统上使用机械连杆进行控制。控制器局域网（CAN）提供了一个简单可靠的通信协议，作为车内网络的标准连接传感器、控制器和互联网等等。随着车对车（V2V）和车对基础设施（V2I）通信接口的出现，采用CAN可以加快应用程序的速度。然而，车辆系统的开放性存在恶意网络攻击的风险，这些攻击很可能严重威胁人类生命。然而，CAN通信一开始是为孤立的系统开发的通信，并未考虑安全性，所以传统的车内网络极易受到网络攻击。例如，每个共享CAN总线的ECU都可以获得任何ECU到ECU的消息。如图1所示，CAN通信报文中不包括发送者的标识，最近的研究工作也指出了CAN通信安全性的不足。

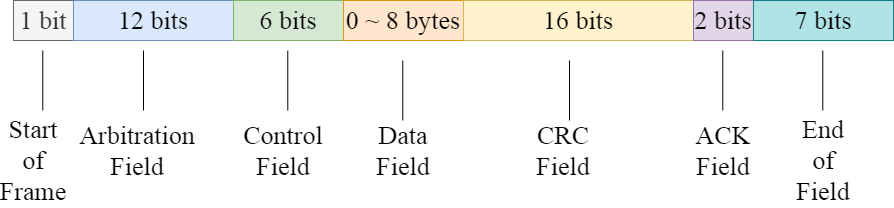


图 1 CAN报文的格式

Koscher等人[[2](#_ENREF_2)]进行实验证明通过数据包注入和修改轻易模糊化CAN消息。文献[[3](#_ENREF_3)]、[[4](#_ENREF_4)]中列举了一些常见的攻击场景，例如禁用制动器、在仪表板上显示错误信息等等。对CAN总线的攻击有多种表现形式。驾驶过程中使用的诊断命令可锁定制动器危及驾驶员人身安全。在正常驾驶过程中不会出现诊断命令，可以容易地检测这类入侵报文，然而检测正常的通信报文中插入的非法报文不能通过简单的检测是否出现诊断报文来做入侵检测。一般来说，越复杂度越高的神经网络可以有更高的识别精度，然而会大大增加入侵报文的检测时间。为了减少车辆CAN通信的入侵检测延迟，采用多目标优化遗传算法同时优化入侵检测模型的准确性和复杂度。

为了应对上述挑战，在本文中，我们提出了一种基于神经演化的深度神经网络入侵检测系统对CAN总线进行入侵检测，如拒绝服务（DoS）和欺骗攻击，具有显著的高准确性和低复杂度。综上所述，本文的主要贡献如下：

1. 利用演化算法分别对报文检测部分的卷积神经网络和图神经网络的结构和超参数进行优化。
2. 利用图神经网络提取消息ID的逻辑序列特征，提出了一种将ID从时域转换为空域的方法。
3. 利用强化学习神经网络动态获取每次检测的CAN数据包以适应CAN报文的动态性和不确定性，使用图神经网络和卷积网络同时提取CAN报文的逻辑和空间特征，并使用公共CAN数据集验证模型，通过进行的大量实验证明提出的检测模型与其它深度学习算法相比具有更好的检测性能。
4. **相关工作**

2.1 车辆CAN 入侵检测的方法

2.1.1 传统方法

CAN通信不支持消息认证码（MAC）[[5](#_ENREF_5)]和其他确保通信安全的方法。一些研究人员尝试创建新的车辆通信协议或者在通信中增加MAC的传输。Tashiro等人[[6](#_ENREF_6)]提出了一种协议，通过在每个帧中发送部分MAC对单个帧和整个部分进行篡改检测，从而可以防止重播、伪装和注入攻击。Nowdehi等人[[7](#_ENREF_7)]从行业角度，根据潜在CAN消息认证解决方案的五个标准研究了许多经过修改的协议。他们发现没有符合所有标准的解决方案。VatiCAN利用维护和支持的方法、“充分的实现细节”和减少多余开销。WooAuth修改了扩展CAN协议，为认证代码提供了更多空间。很多学者认为CAN总线可能“根本不适合安全通信”[[7](#_ENREF_7)]

2.1.2 深度学习方法

入侵检测系统（IDS）可以结合深度学习来训练自己识别异常行为，可以作为MAC的替代或补充。IDS可以防止欺骗、注入、总线关闭和拒绝服务攻击。Choi[[8](#_ENREF_8)]等人介绍了一种利用电压ID的方法，该方法利用ECU信号的不一致性进行训练和测试从而识别信号特征，然后使用训练数据验证ECU是否已损坏。电压入侵检测系统可以使用多类分类器检测伪装攻击，其中一类分类器对应于ECU，它预测最可能的发件人，并将此信息与消息的实际CAN ID进行比较，如果它们不同，则会检测为伪装攻击。song等人[[9](#_ENREF_9)]将CAN帧的ID部分转换为二进制。利用修改后的ResNet模型提取二进制文本的特征，学习入侵消息和正常消息的CANID的二进制空间特征进行入侵检测。实验结果表明，与传统的机器学习算法相比，该算法具有较低的误报率和误报率。Taylor等人[[10](#_ENREF_10)]提出使用深度学习方法进行入侵检测，因为它们直接从网络上的比特流生成，这些功能的执行效率高，复杂性低。该技术在离线训练特征的同时监控车辆网络中的交换数据包，并在实验中以显著的高检测率提供对攻击的实时响应。

2.2 演化算法

2.3 演化算法搜索神经网络架构

神经网络结构搜索（NAS）旨在自动设计网络体系结构，这本质上是一个在资源受限的特定搜索空间中找到性能最佳的网络结构的优化问题[[11](#_ENREF_11)]、[[12](#_ENREF_12)]。Sun等人[[13](#_ENREF_13)]使用具有可变编码长度的EA来自动演化CNN的体系结构。William等人[[14](#_ENREF_14)]介绍了一种基于DAG的进化NAS编码策略，该策略的性能优于随机生成的CNN架构。Real等人提出了Amoebanet[[15](#_ENREF_15)]，它使用改进的锦标赛选择来进化网络群，并且和手工设计的模型对比，在ImageNet上取得了更好的结果。Wang等人[[16](#_ENREF_16)]设计了一种有效的进化算法，以在GANs框架内优化生成器。该方法能有效提高GAN模型的生成性能和训练稳定性。Sun等人[[13](#_ENREF_13)]提出了一种可变长度编码，可以表示不同数量的构建块和不同数量的层，以搜索最佳的深度卷积神经网络。NEMO[45]使用进化多目标方法设计CNN架构，它使用NSGA-II最大化分类性能并最小化网络的推理时间。Elsken等人[[17](#_ENREF_17)]将NAS描述为一个双目标优化问题，其中两个目标是性能最大化和计算资源最小化。Lu等人[[18](#_ENREF_18)]提出了NSGANet，它可以自动设计网络，最大限度地提高模型性能，同时最小化浮点数操作（FLOPs）。

2.2.2 代理模型

EvoNAS的一个主要缺点是在进化优化过程中，每个新的候选神经网络都需要在训练数据集上进行训练，然后在验证数据集上进行评估，避免过度拟合。因此，如果网络庞大，训练数据集庞大，EvoNAS中的架构评估需要非常长的时间。由于EAs是一类基于群体的搜索方法，它们通常需要大量的适应度评估，这使得EvoNAS在计算上难以实现。例如，在CIFAR10和CIFAR100数据集上，CNN-GA[[19](#_ENREF_19)]分别消耗35 GPU天和40 GPU天，演化CNN方法[[20](#_ENREF_20)]消耗17 GPU天，大规模进化算法[[21](#_ENREF_21)]消耗2750 GPU天。因此，在计算资源有限的情况下，代理模型可以加速EvoNAS中的适应度评估。代理被分为高层次代理和低层次代理。高层次代理和低层次代理分别表示架构层次和架构中的参数层次。高层次代理表示通过对神经网络架构的参数化来预测不同神经网络的精度。然而低层次代理解决的是在搜索到多个架构之后，需要每个架构都要从头开始使用SGD优化从而作出评估的复杂性。低层次代理是给定一个训练好的包含所有子架构的超网和神经网络的结构，神经网络架构的权重继承来自超网的权重，在搜索过程中，使用从超网继承的权重的准确性成为选择架构的标准。但是通过权重共享预测架构的准确性和神经架构最终的准确性之间存在的相关性并不是绝对的，我们搜索到的神经架构参考MSuNAS不仅仅只是共享超网的权重，并且再次通过训练进行微调。MetaQNN[[22](#_ENREF_22)]使用代理模型从SGD训练的前25%学习曲线预测候选体系结构的最终精度（作为时间序列预测）。PNAS[[23](#_ENREF_23)]使用替代模型来预测结构的精度，在单元结构中添加了一个额外的分支，这些分支重复堆叠在一起。这两种方法都使用代理的方法来评价神经架构的性能，然而这种方法的预测精度和模型实际精度的相关性是比较低的。OnceForAll[[24](#_ENREF_24)]还使用代理模型来预测架构编码的准确性。然而，代理模型是针对整个搜索空间进行离线训练的，因此需要大量样本进行学习。ChamNet[[25](#_ENREF_25)] 通过完全的低级别优化来训练许多体系结构，并仅选择300个具有不同效率（触发器、延迟、能量）的高精度样本来离线训练替代模型。我们的模型仅在接近帕累托前沿的样本上进行在线学习，显著提升了架构搜索的效率。我们的入侵检测模型评估方法借鉴了MSuNAS的思路。

1. **提出的方法**

我们提出的架构如图2所示，可以分为两个大部分，检测样本采集部分检测部分。样本采集部分强化学习网络根据图神经网络学习到的车辆状态来动态的获取一定长度的CAN报文数据。图网络和卷积网络的网络架构都使用了遗传算法进行了优化。遗传算法优化了卷积网络的卷积层，图神经网络的图卷积层以及全连接层，这些部分在图2中用虚线标出。模型同时利用图神经网络提取逻辑特征的优势，卷积神经网络提取空间特征的优势，最后整合两个网络各自的输出结果得出模型的检测结果，具体过程见下文。

1. 数据采集部分

在每次入侵检测之前先动态采集一定长度的CAN报文数据作为一个输入到检测网络的样本。车辆CAN报文中，不同功能的报文数据，或者是同一功能的报文也可能需要不同的帧数去完成，所以使用强化学习网络解决CAN报文动态采集的问题。在训练强化学习网络之前首先先训练出一个具有入侵检测能力的图神经网络，把图神经网络的卷积层的输出转换成一维向量。模型共有两个图卷积层，会得到两个一维向量，把得到的两个一维向量合并为强化学习的状态输入。根据图网络的识别结果设计奖励函数，如果图网络识别正确获得正向奖励，如果识别错误获得负值的惩罚。强化学习的输出是一个具有多维的向量，维度如下等式1所示。

（1）

入侵检测的最大报文长度与最小报文长度之差。算法强化学习网络架构设计参考TD3[[26](#_ENREF_26)]。

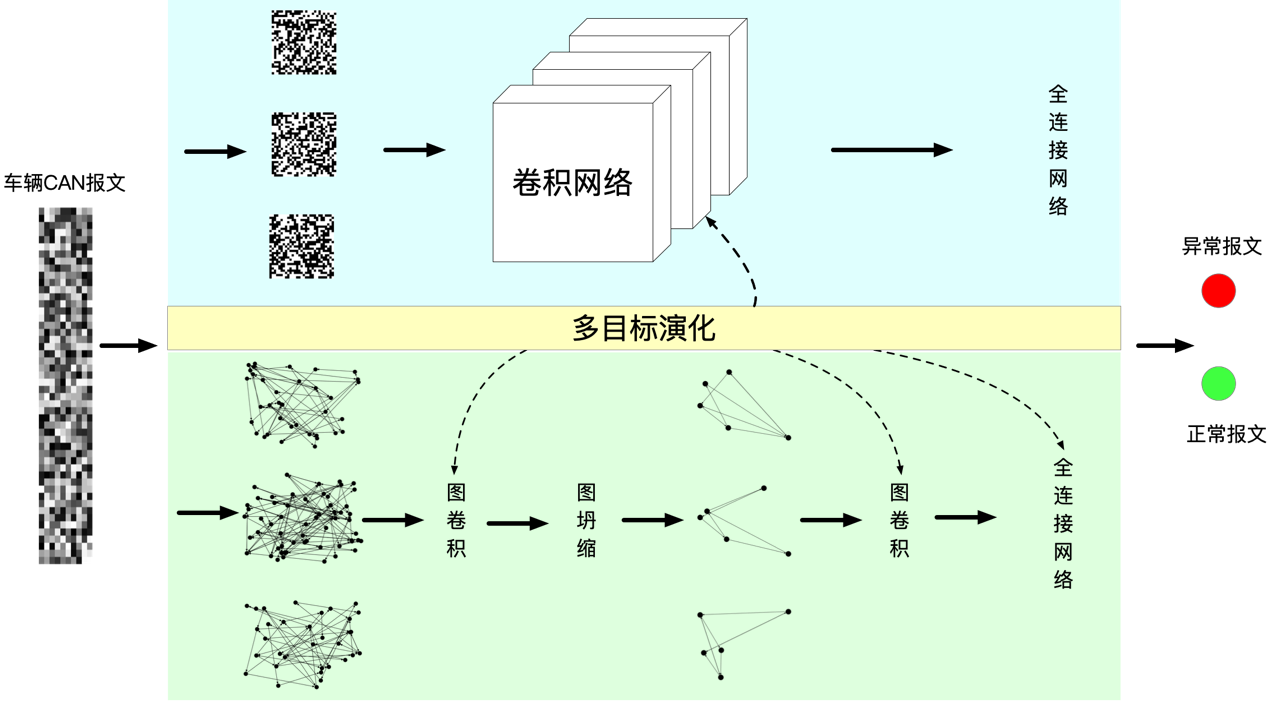


图 2 神经网络的整体架构

2. 图网络构建

2.1构建图数据

模型的训练和测试基于发布在IEEE-DATAPORT上公开数据集攻击与防御挑战2020数据集[[27](#_ENREF_27)]。数据集包含车辆行驶和静止两种状态，每个部分有两种类型的数据集文件：入侵和正常。在入侵文件中入侵CAN报文与普通报文穿插在一起。将这些带标签的消息转换为可输入网络进行训练和验证的格式，模型可以快速准确地识别数据集中的入侵CAN消息。每个数据文件都有数百万个CAN数据帧足够作为本工作提出模型所需要的数据集。

首先，CAN报文中前一时刻报文帧CANID指向下一帧CANID。CANID的种类有限，并非所有CANID都指向新的CANID，它可能指向以前出现过CANID。转换的图形数据的方向关系与消息帧序列的时间序列相关。不同的CANID对应车辆中的不同功能，可以通过图形神经网络学习车辆功能执行的逻辑是否合理。例如，制动时会出现较大的加速信号，或者在没有车钥匙信号的情况下会出现车辆点火信号等不合理的逻辑。学习一定帧数的CAN数据包中的抽象判断消息是否是入侵报文。图数据的构建方法如图3所示。

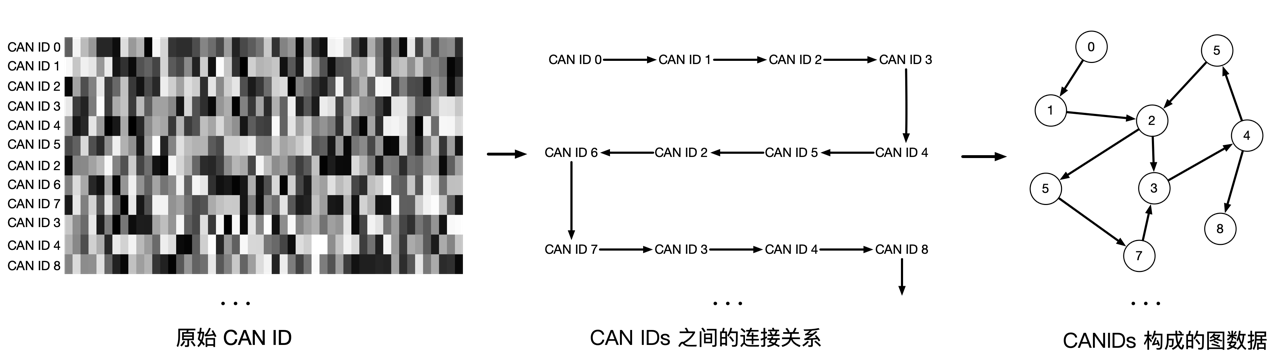


图 3 图数据的构建方法

2.2 整体图网络架构

本工作的图神经网络是图层面的分类任务，图分类任务的网络架构参考[[28](#_ENREF_28)]。 在一些其他采用池化方法的图分类任务是使用求和或者求平均的方法，无差别对待子图中的每个节点，无法提取每个子图中节点的结构信息。本模型的图坍缩算法将构建好的图数据使用聚类的方法分成若干个簇，每个簇可以看作是图数据的子图，得到一系列子图的特征矩阵，使用特征向量构建池化矩阵，把每个子图池化为一个超级节点。

给定k个子图连接的图G，**c**是图G的一部分。表示子图的节点数量，是子图的节点列表。每个子图可以看作是图G的超级节点。定义采样算子如下：

当且仅当

表示矩阵中第i行第j列的元素，表示节点列表中第j个节点。该操作表明节点在子图和在原始图G的对应关系。

因为傅立叶变换可以将图信号转换到频域，可以将信号信息和图数据的结构信息都考虑在内，所以我们参考了【图坍缩】使用傅立叶转换设计池化操作。将构建的图信号G池化为。池化操作基于子图【---公式---】的傅立叶变换。子图的拉普拉斯矩阵为。,…表示第k个子图的拉普拉斯矩阵的特征向量。使用上采样操作上采样特征向量到整个图G。

集合所有上采样的特征向量构成一个矩阵，表示包含所有子图的第*l*个特征向量的池化矩阵。

每个子图不一定有相同的节点数，也就是说每个子图的特征向量的个数是不一定相等的。表示所有子图中最大的节点个数。对于拥有个节点的子图的第l次池化操作表示为：

表示第l次池化操作的结果，的第k行包含的第k个子图的信息，即第k个超级节点。

基于以上结构我们可以构造一系列次的池化操作，结合所有池化操作的结果构成矩阵。

我们的工作对图数据进行了一次池化，池化前和池化后都做了一次图卷积操作，最后把两次图卷积的输出都通过求和或者求平均的方法合并，合并结果输入到全连接层分类。合并两个卷积层输出向量的具体方法使用遗传算法搜索得到。

1.3 使用遗传算法优化图网络

演化算法优化为了得到在保证入侵检测模型检测精度的同时尽可能低的模型复杂度。染色体的第一个基因表示是否使用图卷积的方向信息。虽然使用构建图数据的方向，会更多的利用图数据的信息，实验结果表明使用更多的图信息并不一定可以提高识别精度。根据上一节的图坍缩过程可知，需要得到每个子图的拉普拉斯矩阵之后得出每个子图的特征向量。染色体的第二个基因表示在求每个子图的特征向量之前是否对拉普拉斯矩阵使用正则化。图网络正则化的拉普拉斯矩阵和非正则化的L的公式表示为：

其中A表示图数据的邻接矩阵，D表示图数据的度矩阵。

染色体的第二和第三个位置的基因分别是两个图卷积块的深度，染色体第四和第五个位置的基因预测层中每一层的神经元的数目。在演化算法搜索过程中，动态更改神经网络架构的程序设计参考[[29](#_ENREF_29)]。神经网络图卷积的层数和预测层的神经元个数调整到合适的数量可以得到较低的复杂度达到效果很好的检测准确性。染色体的第六个基因开始一直到最后的所有基因都是神经网络结构中的超参数。染色体中的第9到10位两位分别表示每个卷积块输出卷积结果之后如何降维成一维向量。其中有对每个维度求和，取平均，取最大三种选择。在图网络中的结果设计比如是否使用有向图的信息，构造池化矩阵时是否正则化，每次图卷积的层数，预测层的层数以及预测层中每层的神经元数等等。还有超参数我们也进行了搜索，比如Dropout rate，学习率。具体的基因和相应的对应关系如表1所示。

表 1 图神经网络搜索空间

|  |  |  |
| --- | --- | --- |
| 基因位置 | 表示的含义 | 搜索空间 |
| 0 | 是否使用方向信息 | 有向/无向 |
| 1 | 坍缩图是否正则化 | 正则化/不正则化 |
| 2-3 | 图卷积的层数 | 1 2 3 |
| 4-5 | 预测层每层的神经元数 | 0.25 0.75 1.00 |
| 6 | DropOut | 0.05 0.1 0.2 0.3  0.4 0.5 |
| 7 | 权重衰减 | 5e-4 8e-4 1e-3 4e-3 |
| 8 | 学习率 | 0: 5e-4 1: 1e-3 2: 5e-3 3: 1e-2 |
| 9-10 | 合并图卷积输出的特征向量 | 求和 平均 取最大 |
| 11-18 | 激活函数 | Sigmoid tanh relu  Leaky\_relu relu6 |

2. 卷积部分

2.1 数据获取

数据获取参考卷积网络识别CAN入侵报文论文，首先把十六进制CAN IDs 转化为二进制，构成与图像相似的样本。每个像素点分别是0或者1，参考论文，车辆CAN扩展帧的ID位数是29位，采集29帧构成2929的样本。

2.2 卷积网络架构

搜索空间对于架构设计质量和搜索效率都有很大的影响，在本节介绍搜索空间的基本设计思路。深度卷积网络例如Inception和Resnet的变种都是通过堆叠多个块来设计构建神经网络。网络结构设计包括深度（层数）、宽度（通道数）和空间分辨率变化（池层数）的确定，而块结构设计则规定了分层连接和局部计算。通过这种分块设计方法，生成的模型不仅可以实现高性能，而且可以推广到不同的数据集和任务。因此，我们遵循与[[30-32](#_ENREF_30)]中相同的块级设计方法。块是一个小的卷积网络，为了在前向传播中更有效地处理不同的中间信息，根据特征映射的不同网格大小设计了四种种卷积块。同时，reduction块的设计目的是增加深层的感受野，并通过以2的步幅应用所有操作，将特征图的网格大小减半。按照目前流行的CNN架构[[33-35](#_ENREF_33)]的惯例，当特征图的网格大小减半时，我们将块的通道（过滤器）数量增加一倍，以保持大致恒定的隐藏状态维度。

定义一个操作空间，由一组可能的网络架构基本组件、由人类专家设计的已知成功模块组成。本研究中使用的5种操作和相应的基因型-表型映射如表2所示。在表中，空间可分离卷积(SP)和深度可分离卷积（DW）能够在不牺牲网络性能的情况下减少网络参数。这里我们使用两个DW操作和两个SP操作，内核大小分别为3×3和5×5，简称SP3、DW3和SP5、DW5。

表 2 卷积神经网络的基因编码

|  |  |  |  |
| --- | --- | --- | --- |
| 卷积类型 | 卷积核大小 | 简称 | 编码 |
| 空间可分离卷积 | 3 | SP3 | 0 |
| 空间可分离卷积 | 5 | SP5 | 1 |
| 深度可分离卷积 | 3 | DW3 | 2 |
| 深度可分离卷积 | 5 | DW5 | 3 |
| 常规卷积 | 3 | 3\*3 | 4 |

我们定义遗传算法的交叉操作是在两个亲代染色体上分别随机选择一个点断开，相互交换染色体基因。定义遗传算法的变异操作是在一个染色体上随机选取两个基因，然后进行交换。因为每次都重新训练生成的个体，所需要的计算量是非常巨大的。为了提高遗传算法的评估效率，在交叉操作和变异操作的过程中，把相应神经网络位置的权重参数也进行交换。种群中每个个体经过交叉、变异操作后，会进行训练来评估个体的适应性。为了缩短个体评估的时间，在每一个epoch训练中，会先在种群中随机选出1个个体训练25次，然后再从种群中随机选择个体进行训练，这样可以在较少的epoch训练到种群中的每个个体。通过四个不同的卷积块的示意图可以得知，不同卷积块的卷积操作通道数都是不同的，每个卷积块内的卷积操作的通道数都是相同的，这对于交叉操作是没有影响的，但是会影响变异操作。在一个染色体的两个基因交换位置，并且需要交换神经网络的相应的权重，所以两个基因的通道数需要相同。redA和resB卷积块都只有一个基因，resA和redB卷积块各自分别有三个基因。所以变异操作只能在resA块或者redB块内进行。卷积网络中使用演化算法有的结构如图4所示。在网络架构可以演化优化的操作在图中的红色块标出。

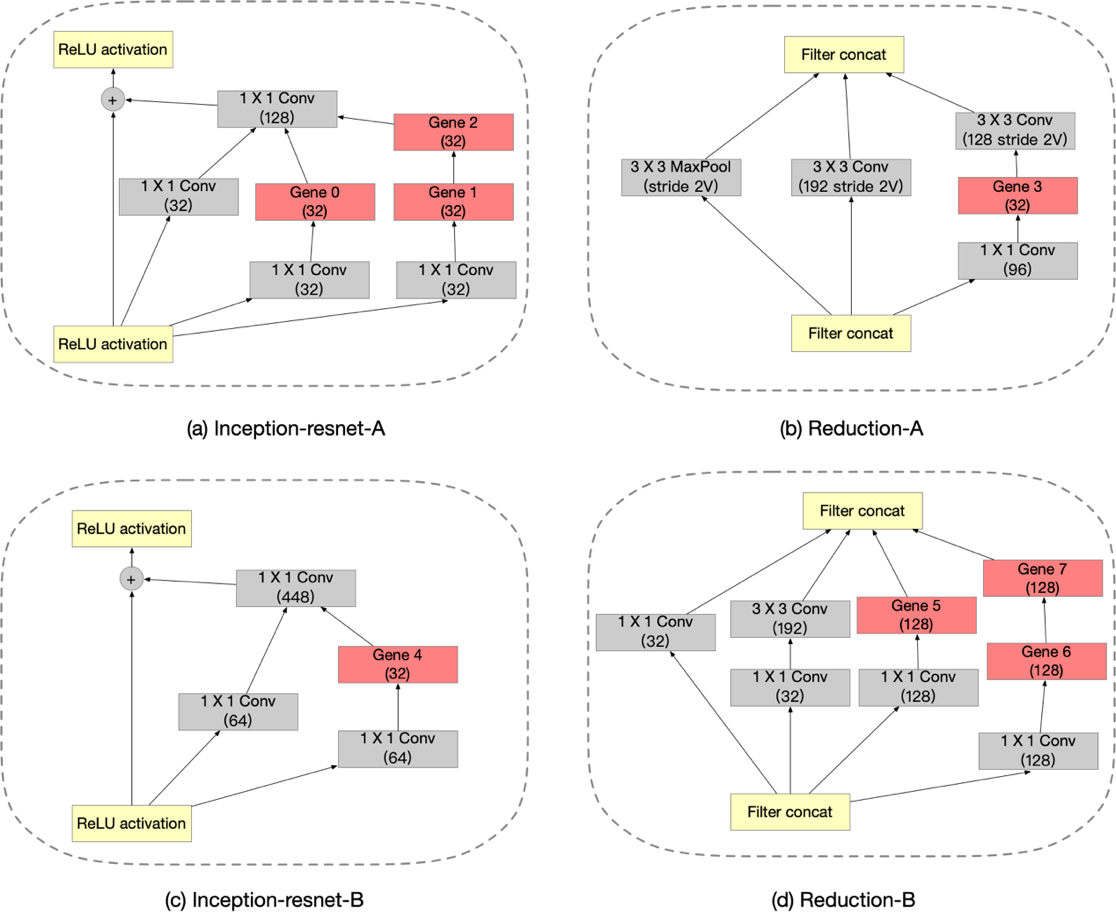


图 4 CNN中可进化的卷积块

Inception-ResNet是深度卷积模型的一种，为了图像分类领域把图片分为1000个类而设计并且表现出了非常出色的性能。神经网络的整体架构如图所示。输入的尺寸是29X29X1经过Stem模块把输入数据尺寸转换为13\*13\*28。经过我们设计的可优化的四个模块后得到的数据尺寸为2\*2\*896。最终经过Softmax模块把数据转化为维度为2的二分类。

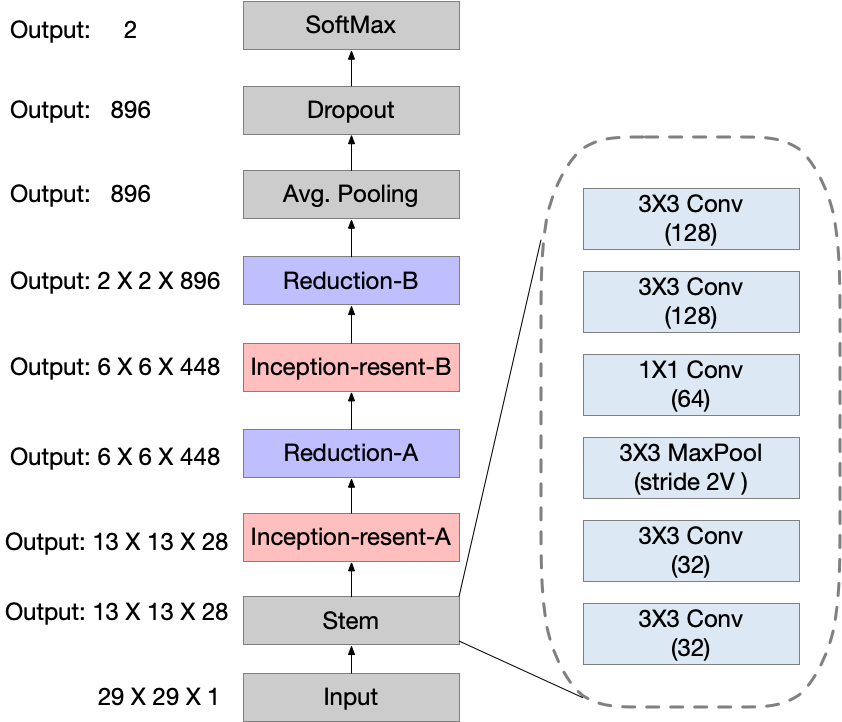


图 5 CNN部分架构

1. **实验和结果**

使用公共数据集Car\_Hacking\_Challenge\_Dataset\_rev20Mar2021测试我们提出的算法框架，并使用了一些其他的深度学习方法进行的对比。数据集分为动态和静态两部分，我们使用数据集中车辆的动态报文部分，使用80%的数据用于训练，使用20%的数据用于测试。我们的网络框架由图神经网络块和卷积网络块两部分组成，所以需要先把数据按照上述的方法先转换为图数据和01二进制数据。在两种网络块之前有强化学习网络负责动态采集输入检测系统的样本。

卷积网络部分预训练26个初代，每个个体训练200个epoch取最好的精度作为适应性指标。然后经过交叉、变异演化30代，每次取最好的26个个体为下一代的亲本，以此类推。在每一代的适应性评估中使用随机选择个体进行训练来提高效率。演化完成后得到30代的精度图像，每一代使用不同的颜色标注出来。从图中可以看出，随着演化过程的进行，精度低的个体越来越少，并且演化出了更高精度的个体。初代的个体的精度基本都在85%左右，一开始的演化，精度范围开始扩大，有的个体的精度达到了50%左右，随着演化过程的进行，个体质量不断上升，直到演化结束后，所有个体都达到了85%以上。如表3所示，尽管演化结束和初代个体的精度相差无几，但是在演化的最后一代选出优良个体经过训练后的识别精度比初代提高5%。

图网络部分，先把数据转换为图数据，经过图坍缩，图卷积，预测层等得到预测结果。图网络的卷积层数，预测层层数，每个预测层的神经元都使用遗传算法进行优化。一开始随机生成100个初代，从初代选取7个帕累托前沿的个体作为下一代的亲代，进化30代。得到30代的复杂度，flops，为x轴，精度误差，acc\_error，为y轴的图像。图中用红色圆点标出了最后三代个体的点。Flops表示网络识别过程中矩阵运算的次数，除以超网的矩阵运算次数，把flops指标化成0到1的范围。Acc\_error指的是 1-acc，所以acc\_error越小，精度越高。从图中可以看到最后个体进化到图中的左下角的位置，代表更高的精度和更低的复杂度。

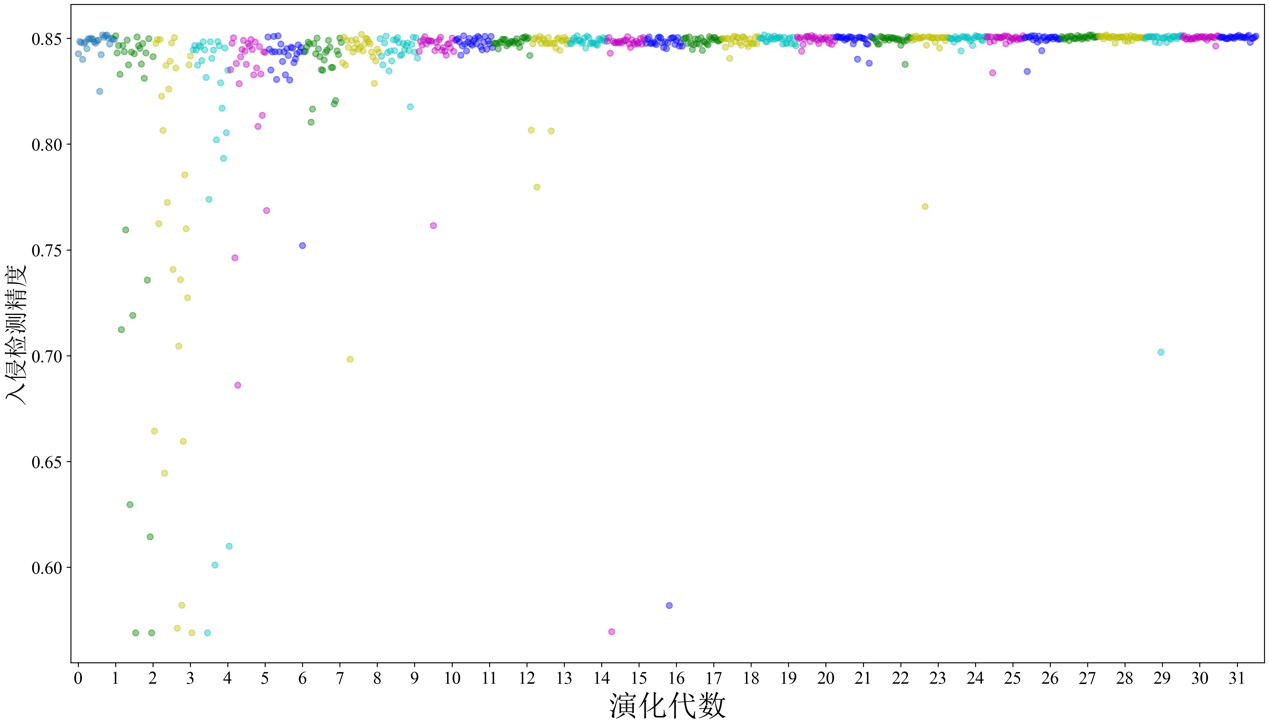


图 6 卷积部分演化结果

表 3 卷积网络优良个体训练结果

|  |  |  |  |
| --- | --- | --- | --- |
| 阶段 | 基因型 | 入侵检测精度 | 复杂度 |
| 初代 | 2, 4, 2, 0, 3, 4, 0, 3 | 85.13% | 2874.5M |
| 演化后未训练 | 4, 4, 4, 2, 3, 0, 2, 3 | 85.15% | 2628.7M |
| 演化后训练后 | 4, 4, 4, 2, 3, 0, 2, 3 | 90.20% | 2628.7M |

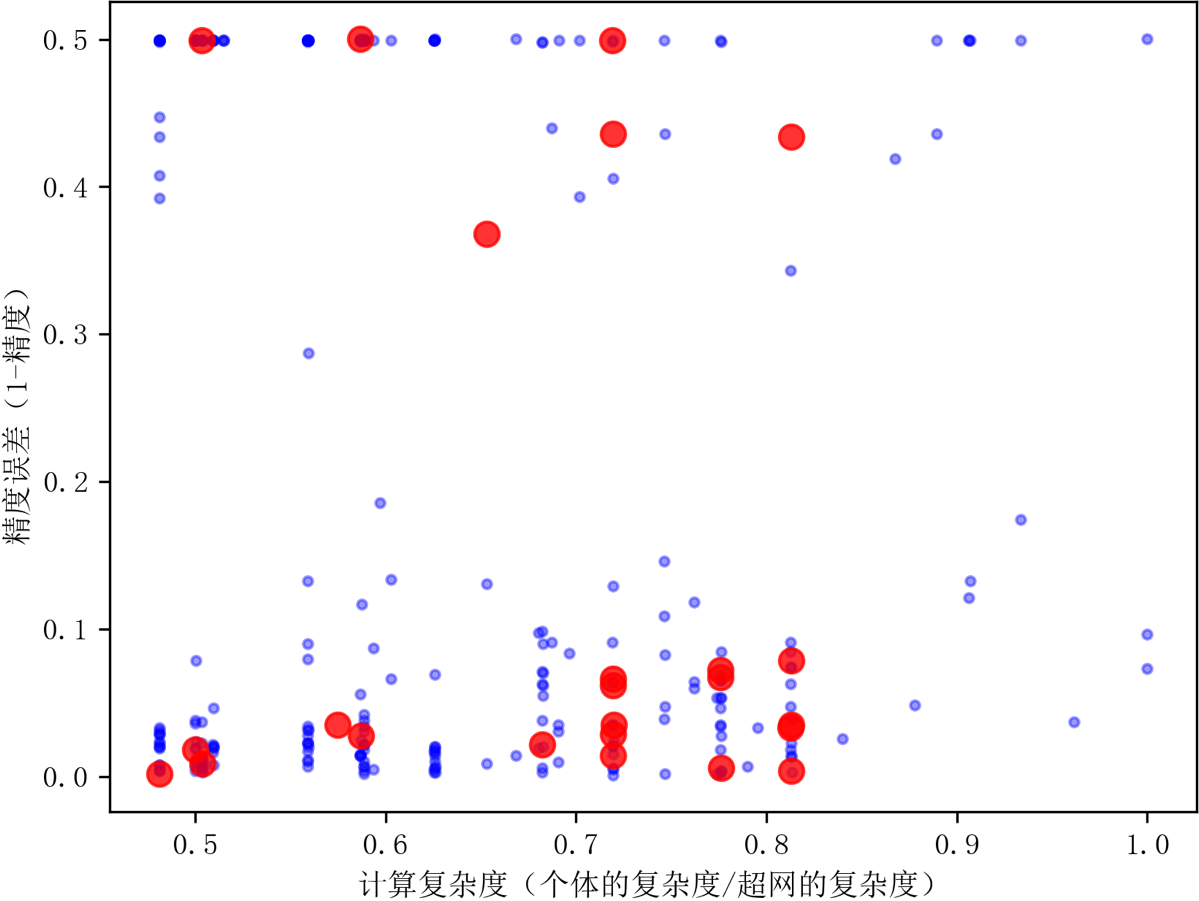


图 7 图卷积部分演化结果

本实验对比实验结果为LSTM和DNN和单独CNN和单独GNN

[[10](#_ENREF_10)]使用LSTM和全连接网络提取CAN的数据部分的时序特征。LSTM提取的是CAN报文里的时序特征，LSTM使用每个位的和下一帧每个位的loss取平均作为整体的loss。在验证时使用取所有位的最大的损失值作为异常信号。我们同样使用了公共数据集Car\_Hacking\_Challenge\_Dataset\_rev20Mar2021进行了验证。使用正常的报文部分训练网络，使用掺有异常报文的数据测试网络。从结果可以看出经过较少的轮数就可以得到不错的结果。在论文中提到可以通过分析某个ID的数据段，一些变化频率很低的位不予考虑。然而该每个CANID都需要单独训练一个完整的神经元网络，所以是比较繁琐的。

[[36](#_ENREF_36)]同样使用了CAN报文的64位的数据段，直接将比特流输入到DNN网络。这个网络更加有效率，可以不做数据处理直接输入到网络。论文中没有指出网络的具体结构，我们采用了5层全连接，第一层64个神经元用来接收64位CAN报文比特流，第二层是128个神经元，第三层是512个神经元，第四层是256层神经元，第五层32个神经元，第六层输出二分类。通过公共数据集的验证，精度较低，但是神经网络简单而且不需要特别的构造数据。

[[9](#_ENREF_9)]和我们的网络卷积块是相同的，使用直接使用卷积部分的网络框架，使用基因444 4 0 444 直接可以构造和论文中一样的架构。它的精度没有我们算法的结果高。

我们的算法集合了卷积网络的空间特征提取和图网络逻辑特征的两个优势，并且在输出结果时比较两个网络输出的二维变量的差值，取差值较大的网络结果为最终结果。也就是取对样本分辨更加自信的的结果。使用这种方法可以让这两种形式的网络相互互补。如表所示，是这几种网络架构的对比结果，可以看到我们的网络可以达到更高的识别精度，这对于网络安全来讲至关重要。

GNN模块也是我们提出的神经网络架构中的一部分，从结果可以看出单独的GNN网络的精度比两个网络结合起来的精度要低。表4中对比了强化学习动态获取CAN数据包对识别精度的影响，引入强化学习增加了整体模型的复杂度，但是可以提高入侵检测精度。

表 4 模型对比结果

|  |  |  |
| --- | --- | --- |
| 深度学习算法 | 精度 | 复杂度 |
| LSTM | 97.02% | 0.057M |
| DNN | 95.36% | 6.8M |
| CNN | 85.06% | 2628.7M |
| GNN | 96.43% | 54.3M |
| 本模型 | 95.17% | 2673.1M |
| 本模型-RL | **99.87%** | 2683.1M |

1. **结论和展望**

我们使用多个网络合成构成在保证入侵检测精度的同时，最大限度减小网络的复杂度。据我们所知，在本工作中第一次使用图网络做CAN入侵检测工作，并且仅仅只使用图神经网络部分检测结果也可以达到95%以上。模型同时融汇了图神经网络和卷积神经网络的双重优点，同时参考两个网络输出的结果，使得两个网络的检测结果形成互补。同时利用CAN数据的逻辑特征和空间特征。在本工作中还有一些不足，虽然可以达到非常好的识别精度，和之前的工作对比，我们的网络的复杂度更高。我们以后的工作可以更进一步应用演化算法更细致的对网络的结构进行精简。

1. 参考文献

[1] G. Leen and D. Heffernan, "Expanding automotive electronic systems," *Computer,* vol. 35, no. 1, pp. 88-93, 2002, doi: 10.1109/2.976923.

[2] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*, 2010: IEEE Computer Society, pp. 447-462, doi: 10.1109/SP.2010.34.

[3] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *USENIX Security Symposium*, 2011, vol. 4, no. 447-462: San Francisco, p. 2021.

[4] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *Def Con,* vol. 21, no. 260-264, pp. 15-31, 2013.

[5] O. Avatefipour and H. Malik, "State-of-the-art survey on in-vehicle network communication (CAN-Bus) security and vulnerabilities," *arXiv preprint arXiv:1802.01725,* 2018.

[6] A. Tashiro, H. Muraoka, S. Araki, K. Kakizaki, and S. Uehara, "A secure protocol consisting of two different security-level message authentications over CAN," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 13-16 Dec. 2017 2017, pp. 1520-1524, doi: 10.1109/CompComm.2017.8322794.

[7] N. Nowdehi, A. Lautenbach, and T. Olovsson, "In-vehicle CAN message authentication: An evaluation based on industrial criteria," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017: IEEE, pp. 1-7.

[8] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "Voltageids: Low-level communication characteristics for automotive intrusion detection system," *IEEE Transactions on Information Forensics and Security,* vol. 13, no. 8, pp. 2114-2129, 2018.

[9] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications,* vol. 21, p. 100198, 2020/01/01/ 2020, doi: <https://doi.org/10.1016/j.vehcom.2019.100198>.

[10] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 17-19 Oct. 2016 2016, pp. 130-139, doi: 10.1109/DSAA.2016.20.

[11] P. Ren *et al.*, "A comprehensive survey of neural architecture search: Challenges and solutions," *arXiv preprint arXiv:2006.02903,* 2020.

[12] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter, "NAS-Bench-101: Towards Reproducible Neural Architecture Search," presented at the Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2019. [Online]. Available: <https://proceedings.mlr.press/v97/ying19a.html>.

[13] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving Deep Convolutional Neural Networks for Image Classification," *IEEE Transactions on Evolutionary Computation,* vol. 24, no. 2, pp. 394-407, 2020, doi: 10.1109/TEVC.2019.2916183.

[14] W. Irwin-Harris, Y. Sun, B. Xue, and M. Zhang, "A Graph-Based Encoding for Evolutionary Convolutional Neural Network Architecture Design," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, 10-13 June 2019 2019, pp. 546-553, doi: 10.1109/CEC.2019.8790093.

[15] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized Evolution for Image Classifier Architecture Search," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 33, no. 01, pp. 4780-4789, 07/17 2019, doi: 10.1609/aaai.v33i01.33014780.

[16] C. Wang, C. Xu, X. Yao, and D. Tao, "Evolutionary Generative Adversarial Networks," *IEEE Transactions on Evolutionary Computation,* vol. 23, no. 6, pp. 921-934, 2019, doi: 10.1109/TEVC.2019.2895748.

[17] T. Elsken, J. H. Metzen, and F. Hutter, "Efficient multi-objective neural architecture search via lamarckian evolution," *arXiv preprint arXiv:1804.09081,* 2018.

[18] Z. Lu *et al.*, "NSGA-Net: neural architecture search using multi-objective genetic algorithm," presented at the Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 2019. [Online]. Available: <https://doi.org/10.1145/3321707.3321729>.

[19] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification," *IEEE Transactions on Cybernetics,* vol. 50, no. 9, pp. 3840-3854, 2020, doi: 10.1109/TCYB.2020.2983860.

[20] L. Xie and A. Yuille, "Genetic cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1379-1388.

[21] E. Real *et al.*, "Large-Scale Evolution of Image Classifiers," presented at the Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2017. [Online]. Available: <https://proceedings.mlr.press/v70/real17a.html>.

[22] B. Baker, O. Gupta, R. Raskar, and N. Naik, "Accelerating neural architecture search using performance prediction," *arXiv preprint arXiv:1705.10823,* 2017.

[23] C. Liu *et al.*, "Progressive neural architecture search," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 19-34.

[24] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: Train one network and specialize it for efficient deployment," *arXiv preprint arXiv:1908.09791,* 2019.

[25] X. Dai *et al.*, "Chamnet: Towards efficient network design through platform-aware model adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11398-11407.

[26] S. Fujimoto, H. Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," presented at the Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2018. [Online]. Available: <https://proceedings.mlr.press/v80/fujimoto18a.html>.

[27] H. Kang, B. I. Kwak, Y. H. Lee, H. Lee, H. Lee, and H. K. Kim, "Car Hacking: Attack & Defense Challenge 2020 Dataset," February 3 2021, doi: <https://dx.doi.org/10.21227/qvr7-n418>. IEEE Dataport.

[28] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, "Graph Convolutional Networks with EigenPooling," presented at the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining, Anchorage, AK, USA, 2019. [Online]. Available: <https://doi.org/10.1145/3292500.3330982>.

[29] J. Yu and T. S. Huang, "Universally slimmable networks and improved training techniques," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1803-1811.

[30] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578,* 2016.

[31] X. Jin *et al.*, "Rc-darts: Resource constrained differentiable architecture search," *arXiv preprint arXiv:1912.12814,* 2019.

[32] X. Zheng, R. Ji, L. Tang, B. Zhang, J. Liu, and Q. Tian, "Multinomial distribution learning for effective neural architecture search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1304-1313.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[34] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.

[36] M. Kang and J. Kang, "A Novel Intrusion Detection Method Using Deep Neural Network for In-Vehicle Network Security," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 15-18 May 2016 2016, pp. 1-5, doi: 10.1109/VTCSpring.2016.7504089.