**基于多目标深度神经演化的车辆CAN报文入侵检测算法**

**摘要**

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，与此同时要确保车辆各模块之间通信的安全性和可靠性，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。基于多目标演化（MOEA）的神经结构搜索（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，通过多目标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输出的特征向量来学习车辆状态来动态地获取每一次检测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。

检测模型的卷积网络和图神经网络使用多目标演化算法优化。卷积网络在演化过程中继承亲本的权重，图网络在演化过程中继承超网的权重，权重共享缩短了个体适应性评估的时间。图网络演化后的flops减小到超网的%40,但是精度却达到95%以上，卷积网络的精度相较于第一代亲本也提高了5%。模型合并两个模型进行入侵检测精度高达99%以上。

**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

1. **介绍**

在过去几十年中，汽车电子产品的应用快速增长[[1](#_ENREF_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[[2](#_ENREF_2)]，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[[3](#_ENREF_3)]。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。Koscher等人[[4](#_ENREF_4)]进行实验证明通过数据包注入和修改轻易模糊CAN消息。文献[[5](#_ENREF_5)]、[[6](#_ENREF_6)]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。【车辆CAN入侵检测工作的必要性】

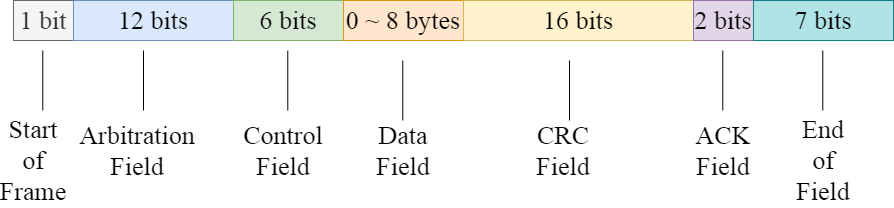


图 1 CAN报文的格式

【图网络构建】图数据无处不在[[7](#_ENREF_7), [8](#_ENREF_8)]，主要用于处理非欧几里德数据特征，包括社交网络[[9](#_ENREF_9)]、文本网络[[10](#_ENREF_10)]和生物网络[[11](#_ENREF_11)]等。图神经网络（GNN）被提议直接对图进行操作，以端到端的方式解决与图相关的问题[[12](#_ENREF_12)]。关系建模对于许多网络或图形数据挖掘任务（如链接预测）至关重要。GNN最近在现实世界中获得了大量成功的应用，如图像识别[[13](#_ENREF_13)]、新药发现[[14](#_ENREF_14)]和流量预测[[15](#_ENREF_15)]等。在本工作通过CAN报文的时序特征构造图数据，并使用图数据图网络提取车辆CAN入侵检测的逻辑特征。现实中存在更多的是动态图，大多数工作都是处理的动态图，比如社交网络或者金融网络。这种动态图是随着时间的推移，之前的图数据的节点或者节点与节点之间的连接发生了变化,这种图网络使用RNN[[16](#_ENREF_16), [17](#_ENREF_17)]学习动态图的时间特征，时空图神经网络结构学习动态图空间特征[[18](#_ENREF_18)]。然而有些图数据有可能在不同的时刻，图是完全不同的。比如本工作构建CAN图数据，两个时间段构造出来的图数据的节点或者连接又可能完全不同。对于识别这种图数据的图网络的工作非常少。本模型使用图坍缩网络[[19](#_ENREF_19)]对图数据进行入侵检测

【网络的神经搜索】为特定场景（例如，生物和物理网络数据）开发由多个GNN层组成的定制学习架构仍然很棘手，即使对于神经网络专家来说也是如此。Falkner等人[[20](#_ENREF_20)]指出，深度学习算法对许多超参数非常敏感。文献[[21](#_ENREF_21)]中的工作表明，GNNs的超参数优化（HPO）对于在实践中取得令人满意的结果至关重要。因此，研究适用于GNN的有效HPO方法对于GNN应用于各种实际问题具有重要价值。为了自动化模型选择过程，神经架构搜索（NAS）被广泛使用[[22](#_ENREF_22), [23](#_ENREF_23)]，近年来已成为深度学习研究的焦点。NAS旨在从搜索空间探索最优的架构组件组合，以最大限度地提高适用于目标问题的模型性能。迄今为止，人们已经付出了巨大的努力来搜索卷积神经网络（CNN）架构，这推动了许多重要基准任务的最新进展，例如，CIFAR10/100和ImageNet上的图像分类[[24](#_ENREF_24), [25](#_ENREF_25)]。相比之下，在图形结构数据的GNN学习方面做的工作很少[[26-28](#_ENREF_26)]。NAS文献提出了两种主要类型的方法作为最有效的‌解决问题的方法：强化学习（RL）和进化算法（EAs）[[29](#_ENREF_29)]。到目前为止，第二种技术在GNNs环境中被忽视了。结果表明，RL和EA都能够在精度方面找到等价的模型，在某些情况下EA速度更快。最近的两项相关工作[[28](#_ENREF_28), [30](#_ENREF_30)]主要关注基于强化学习的NAS，并采用递归神经网络（RNN）作为控制器来生成描述GNN结构的变长字符串以最大限度地提高生成的结构在验证数据集上的预期精度。尽管取得了令人鼓舞的结果，但现有工作始终面临以下两个缺点的挑战：超参数不变和高计算强度。此外，控制器通常生成候选GNN结构，并以顺序方式对其进行评估，这很难扩展到较大的搜索空间。

【演化算法搜索NAS】NAS任务通常可以表述为一个复杂的优化问题[[29](#_ENREF_29), [31](#_ENREF_31)]。一般来说，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。我们的算法使用入侵检测准确率和模型复杂度两个相互矛盾的目标进行多目标优化。在计算智能领域，进化算法（EAs）已广泛用于解决各种神经网络训练问题[[32](#_ENREF_32)]，如权重训练[[32](#_ENREF_32)]、架构设计[[33](#_ENREF_33)]和学习规则自适应[[34](#_ENREF_34)]。最近，采用EA作为NAS优化器的进化神经架构搜索受到了越来越多的关注[[35-37](#_ENREF_35)]。GNN基因[[26](#_ENREF_26)]是通过进化方法提出的，它在两个搜索空间中识别GNN结构和训练超参数，交替优化图网络的结构和超参数（比如学习率，Dropout），我们的工作把它们放在一个搜索空间做搜索，搜索到的图网络架构识别率高达95%以上，图网络和卷积网络合并一起的识别率高达99%以上。

【演化算法-代理-节点继承】神经结构搜索是一个反复尝试的过程，通过迭代生成和评估超参数设置，直到满足预设的停止条件为止，以获得最佳解决方案。通常，选择神经网络的目标函数作为适应度函数来评估超参数。然而，评估通常非常昂贵，因为训练GNN模型需要大量计算资源。另一方面，随着深度学习的发展，神经网络具有越来越多的层或各种其他可选超参数（例如激活函数、优化方法），这导致了较大的搜索空间并增加了搜索的难度。网络组件的增多，给搜索网络需要的计算资源消耗是巨大的，而且为GNN结构调整和优化付出繁琐而艰辛的努力。因此，现有神经结构搜索工作都是基于如何以较少的试验找到最佳解决方案、降低评估成本和提高勘探能力的问题进行的。尽管EAs在各种优化任务上具有竞争性的搜索性能[[38](#_ENREF_38)]，但作为一类基于群体的优化方法，其计算成本通常很高。这对于EvoNAS来说尤其如此，因为EAs通常需要大量的适应度评估，而NAS中的每个适应度评估在计算上都很昂贵，因为它通常涉及在大量数据上从头开始训练深层神经网络。例如，AE-CNN需要27个GPU天才能在CIFAR10数据集上获得优化的CNN体系结构。有一些工作搜索图网络的超参数，通过快速评估个体节省计算资源，比如[[39](#_ENREF_39)]提出并行图结构搜索提高搜索效率，我们的工作中卷积网络演化时子代继承父代网络的权重以及图神经网络是继承超网的权重参数实现快速的个体适应性评估。

【强化学习】深度强化学习是强化学习和深度学习的结合。这一研究领域能够解决以前机器无法完成的一系列复杂决策任务。因此，deep RL在医疗保健、机器人、智能电网、金融等领域开辟了许多新的应用[[40](#_ENREF_40)]。车辆执行不同的功能时的CAN报文时长度和内容有差别的，基于这个启发，我们使用强化学习获取每次测检CAN报文的长度。经过实验结果得到使用变长度的CAN报文样本后模型的入侵检测率提高5%。强化学习网络学习每次图网络中图卷积输出的特征向量当作每次的state输入到强化学习actor网络，强化学习输出下一次的动作，即下一次检测CAN报文的长度。早期的强化学习存在对真实的动作价值的高估问题[[41](#_ENREF_41)]，[[42](#_ENREF_42)]建立两个价值网络解决高估问题。强化学习按照输出动作的类型分为离散型和连续型。模型中动作的含义是输出CAN报文的长度，我们给出报文长度的范围，最小和最大长度之间的整数值都是可以取的。综合以上需求，我们使用td3[[43](#_ENREF_43)]网络作为动态获取CAN报文长度的神经网络。

为了应对上述挑战，在本文中，我们提出了一种基于多目标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. ~~提出多目标演化算法分别对CAN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. 提出一种新的图神经网络检测模型，并将检测准确率和模型flops作为目标使用多目标演化算法进行优化。并且在演化过程中采用节点继承策略以缩短个体评估的时间。
3. 提出一种CAN报文数据转换图数据的方法，根据相邻两个时间戳的时序信息构建成图数据的两个彼此连接的节点。提取CAN报文中的逻辑特征，从而可以对CAN报文进行图分类检测。
4. 提出图网络和卷积网络相结合检测CAN入侵报文，并采用强化学习网络动态采集CAN入侵检测样本。在检测模型训练时同时学习CAN报文的逻辑特征和空间特征，实验结果显示获得了更好的检测效果。
5. **相关工作**

**在本节我们简明介绍车辆CAN报文入侵检测，多目标演化神经网络的相关工作。**

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）[[44](#_ENREF_44)]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[[45](#_ENREF_45)]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[[46](#_ENREF_46)]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”[[46](#_ENREF_46)]

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[[47](#_ENREF_47)]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[[48](#_ENREF_48)]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[[49](#_ENREF_49)]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

2.2.1 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[[50](#_ENREF_50)]、[[51](#_ENREF_51)]。Sun等人[[52](#_ENREF_52)]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[[53](#_ENREF_53)]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[[54](#_ENREF_54)]，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[[55](#_ENREF_55)]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。Yin[[56](#_ENREF_56)]使用进化多目标方法设计CNN架构，它使用基于概率的SMBO最大化分类性能并最小化网络的推理时间。Elsken等人[[57](#_ENREF_57)]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[[58](#_ENREF_58)]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[[59](#_ENREF_59)]分别消耗35 GPU天和40 GPU天，演化CNN方法[[60](#_ENREF_60)]消耗17 GPU天，大规模进化算法[[61](#_ENREF_61)]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。模型搜索神经架构的方法参考MSuNAS[[62](#_ENREF_62)]，该算法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[[63](#_ENREF_63)]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[[64](#_ENREF_64)]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[[65](#_ENREF_65)]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[[66](#_ENREF_66)] 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. **架构搜索空间**

在本节介绍模型的搜索空间，包括图神经网络的搜索空间，卷积网络搜索空间。图神经网络的搜索包括结构搜索和超参数搜索。结构搜索包括图卷积层的层数，图网络部分预测层的宽度等。超参数搜索包括学习率，激活函数类型，Dropout率等等。

卷积网络是参考卷积网络检测CAN报文的工作的卷积网络架构[[48](#_ENREF_48)]，他的架构是自己手工设计的，由一个Stem部分和4个Res卷积块部分组成，我们的模型的卷积部分保留了Stem部分，在4个卷积块上挖取了8个位置。这8个位置中的每个位置有5种选择。

**3.1 图网络部分**

具体的基因和相应的对应关系如表1所示。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，需要从每个子图的拉普拉斯矩阵中得出每个子图的特征向量，这些特征向量构成的矩阵组成坍缩池化矩阵。是否对拉普拉斯矩阵正则化影响图的池化结果。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，动态更改神经网络结构的程序设计参考[[62](#_ENREF_62), [67](#_ENREF_67)]。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。染色体第9部分代表卷积或完全连接层结束后的激活功能。

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 0.50 0.75 1.00 |
| 6 | DropOut | 0.005 0.01 0.02 0.03  0.04 0.05 |
| 7 | 权重衰减 | 5e-4 8e-4 1e-3 4e-3 |
| 8 | 学习率 | 5e-4 1e-3 5e-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 |

**3.2 卷积网络部分**

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。

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 |

1. **提出的方法**

我们提出的架构如图2所示，可以分为检测样本采集部分和检测部分两个大部分。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. 强化学习数据采集部分

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。图神经网络共有两个图卷积层，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。强化学习的输出下一次检测CAN报文的长度，所以强化学习的数据是一个离散的多维向量，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。强化学习网络架构设计参考TD3[[43](#_ENREF_43)]。

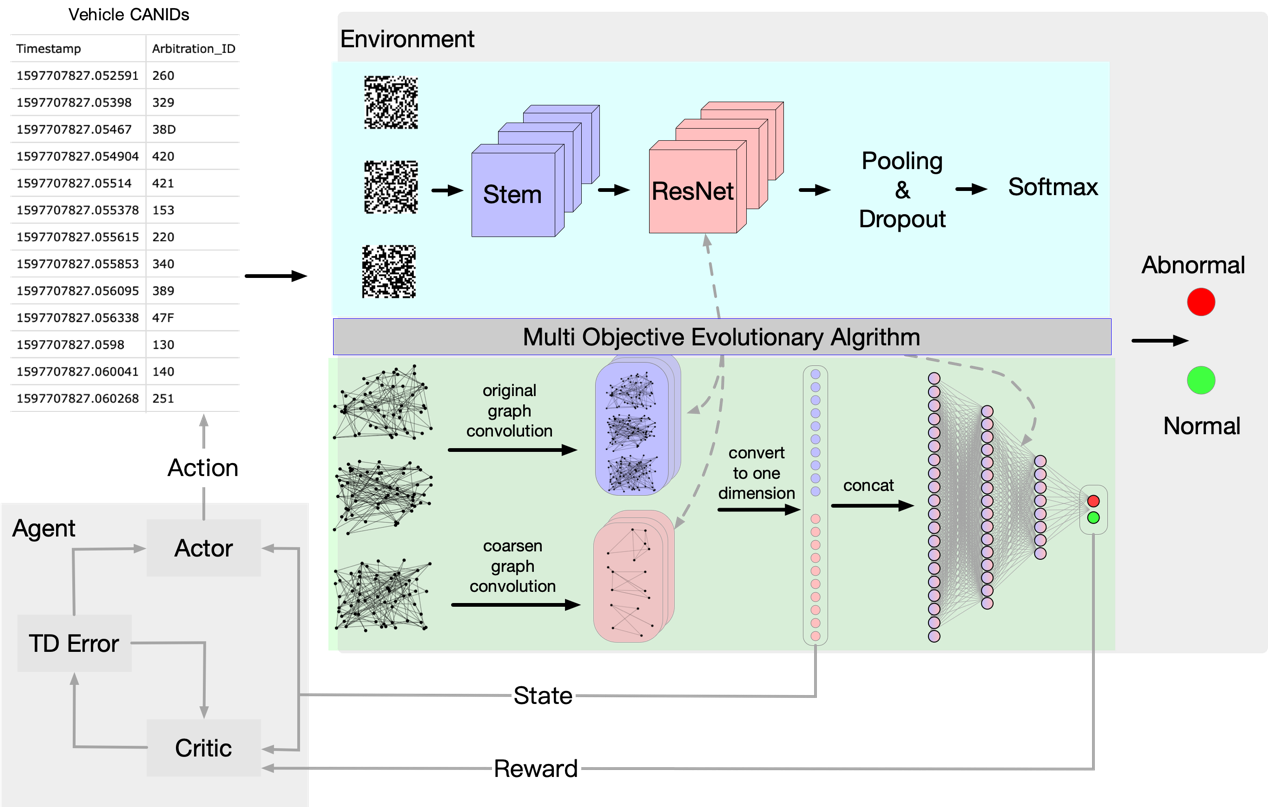


图 2 神经网络的整体架构

2. 图网络部分

2.1构建图数据

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8所示。

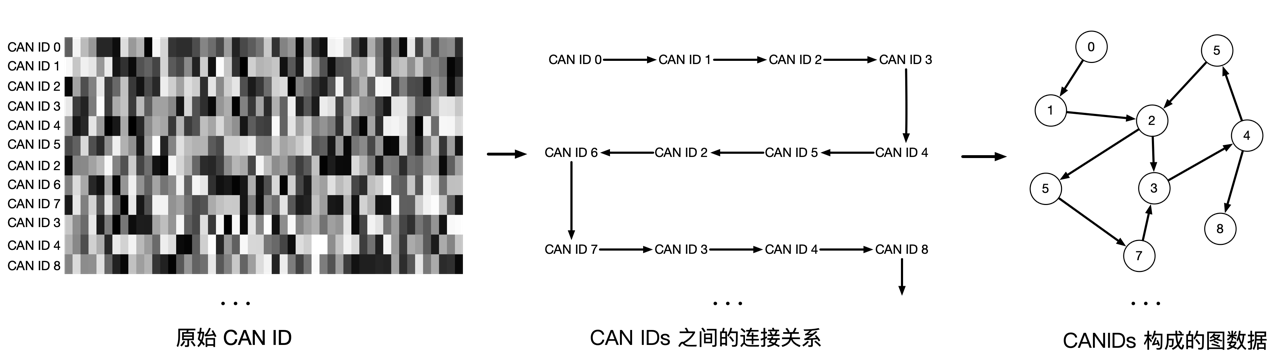


图 3 图数据的构建方法

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[[19](#_ENREF_19)]。 图网络部分的主要想法是通过图卷积对原始图和池化后的图分别做图卷积，将两次图卷积的结果转换为两个一维向量，合并两个一维向量送入全连接层完成检测任务。 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[[19](#_ENREF_19)]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。

2.3图网络的演化过程

图网络的交叉操作和变异操作如图。交叉操作是首先随机在种群中选出两个亲本，然后分别在每个亲本上随机选择交叉点，分开的部分两两互换生成两个新的子代个体。变异操作是对每一个亲本以变异概率p进行变异，再随机在亲本上选择变异点发生基因突变。

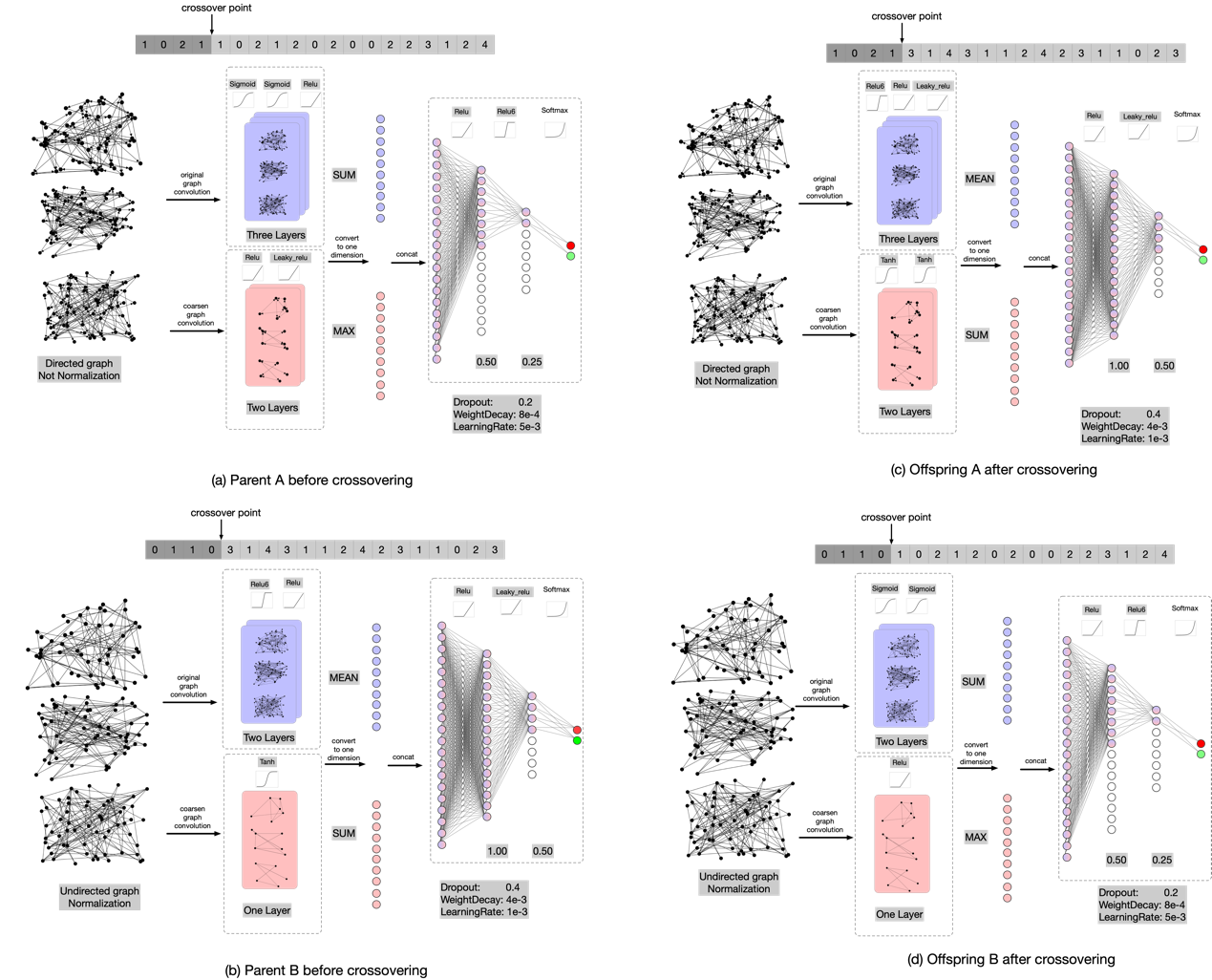


图 4 GNN交叉过程

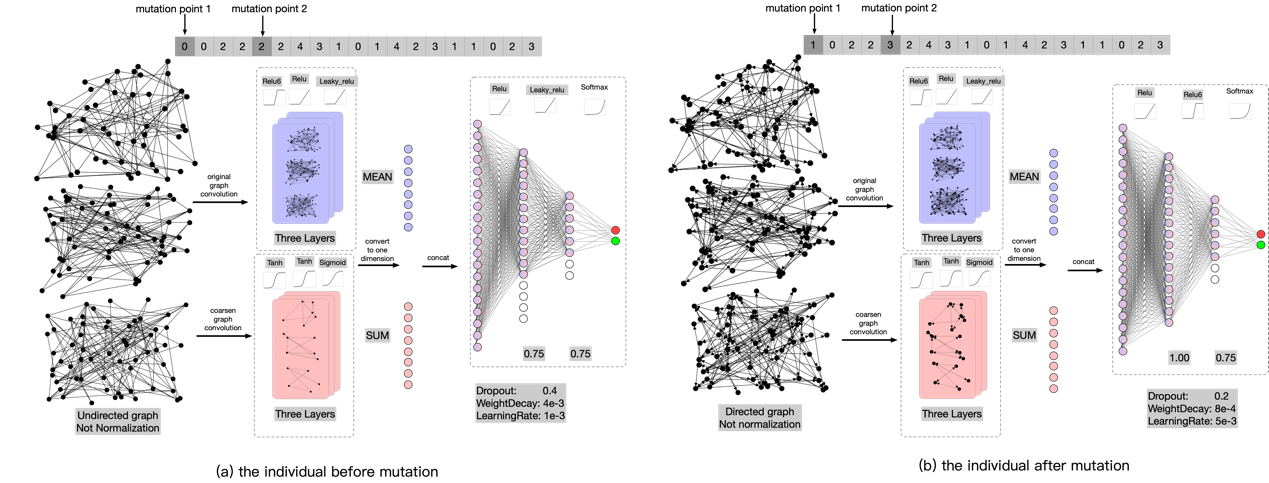


图 5 GNN变异过程

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[[48](#_ENREF_48)]，车辆CAN扩展帧的ID位数是29位，采集29帧构成2929的样本。通过对样本的处理把检测任务转换为类似于图像分类任务。

2.2 卷积网络架构

因此，我们遵循与[[68-70](#_ENREF_68)]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[[71-73](#_ENREF_71)]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。

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6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。

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。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。

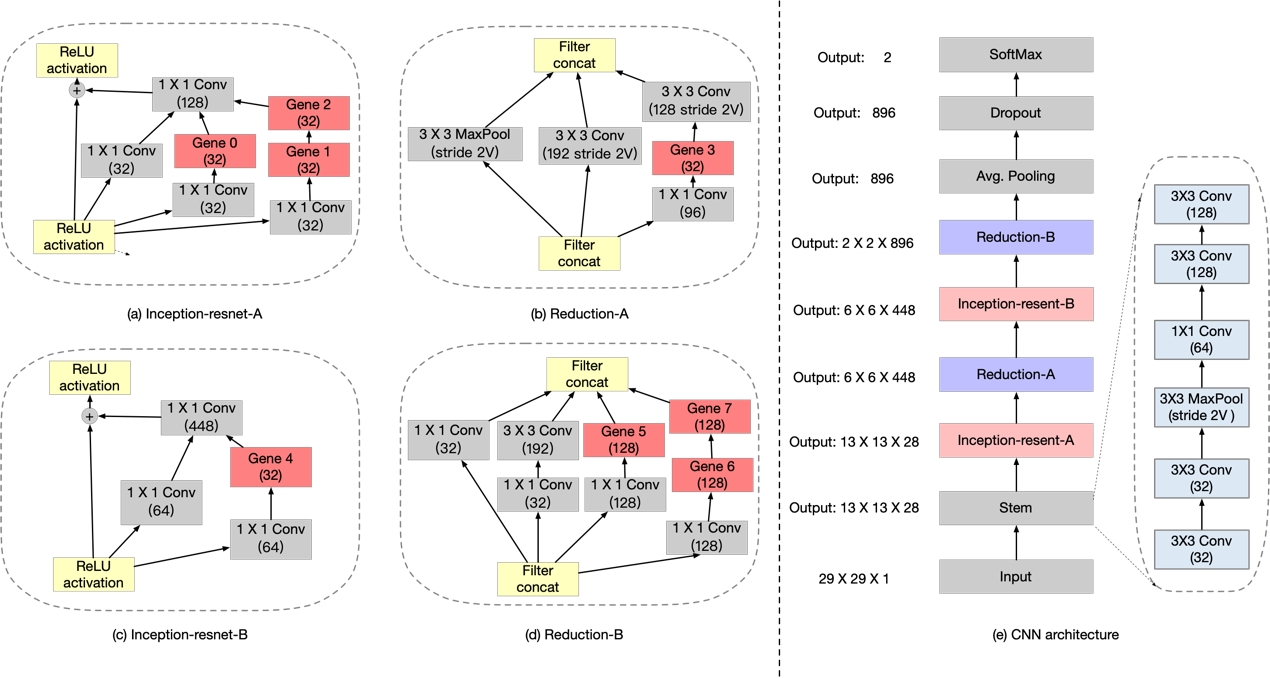


图 6 CNN整体架构以及可进化的卷积块

2.3 卷积网络的演化过程

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所需要卷积核的总的通道数都是不同的，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。

演化部分的交叉操作首先n=2的锦标赛法取出两个亲本个体，然后分别在每个亲本上随机选择交叉点，在交叉点处断开，分开的部分两两互换生成两个新的子代个体。变异操作是对每一个亲本以变异概率p进行变异，再随机在亲本上选择两个点发生交换。

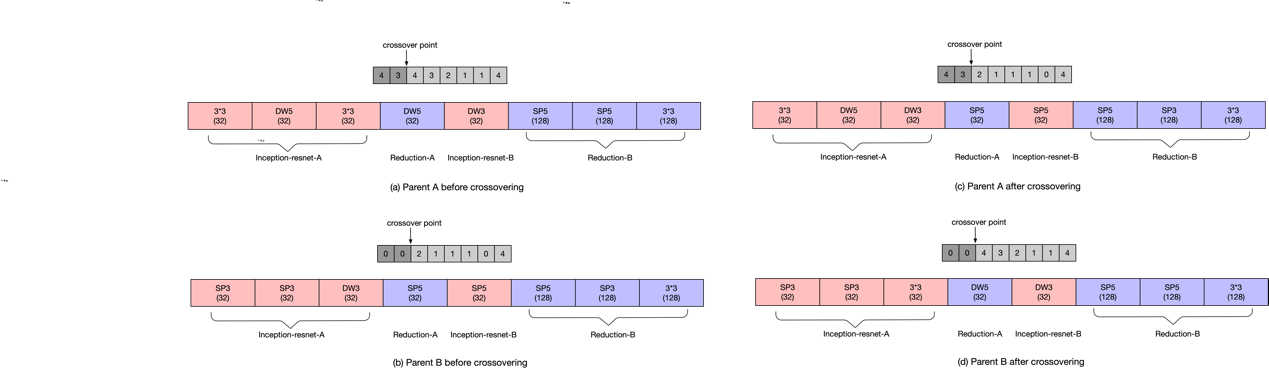
****

图 7 CNN交叉过程

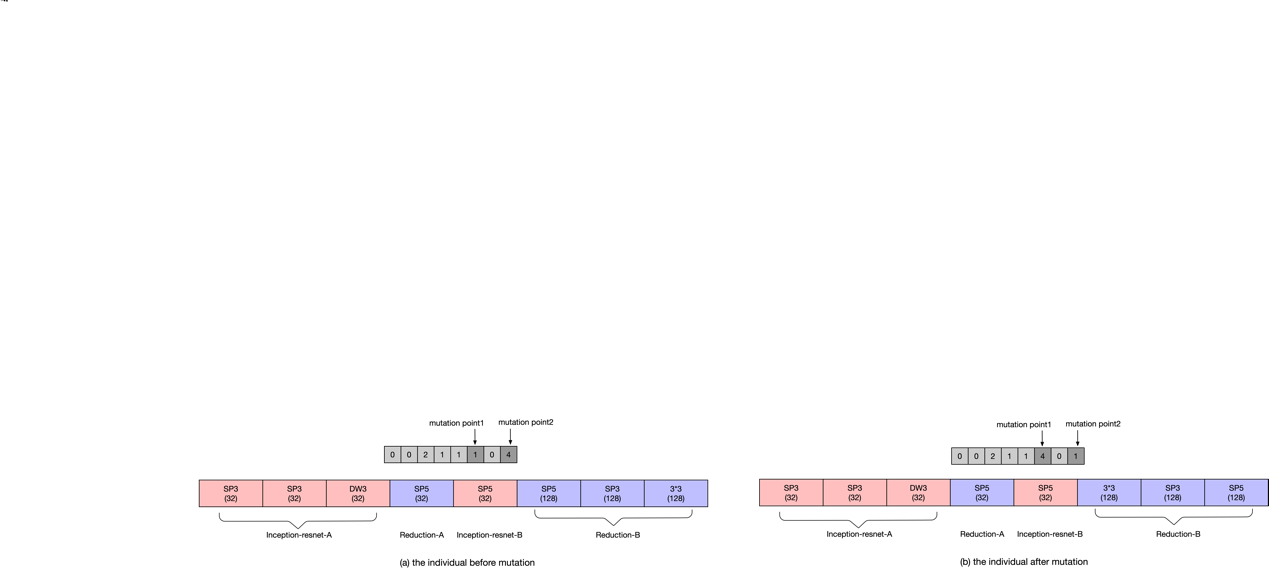
****

图 8 CNN变异过程

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。

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[[74](#_ENREF_74)]。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。

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9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。如图10所示，得到30代的复杂度，flops，为x轴，精度误差，acc\_error，为y轴的图像。10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。

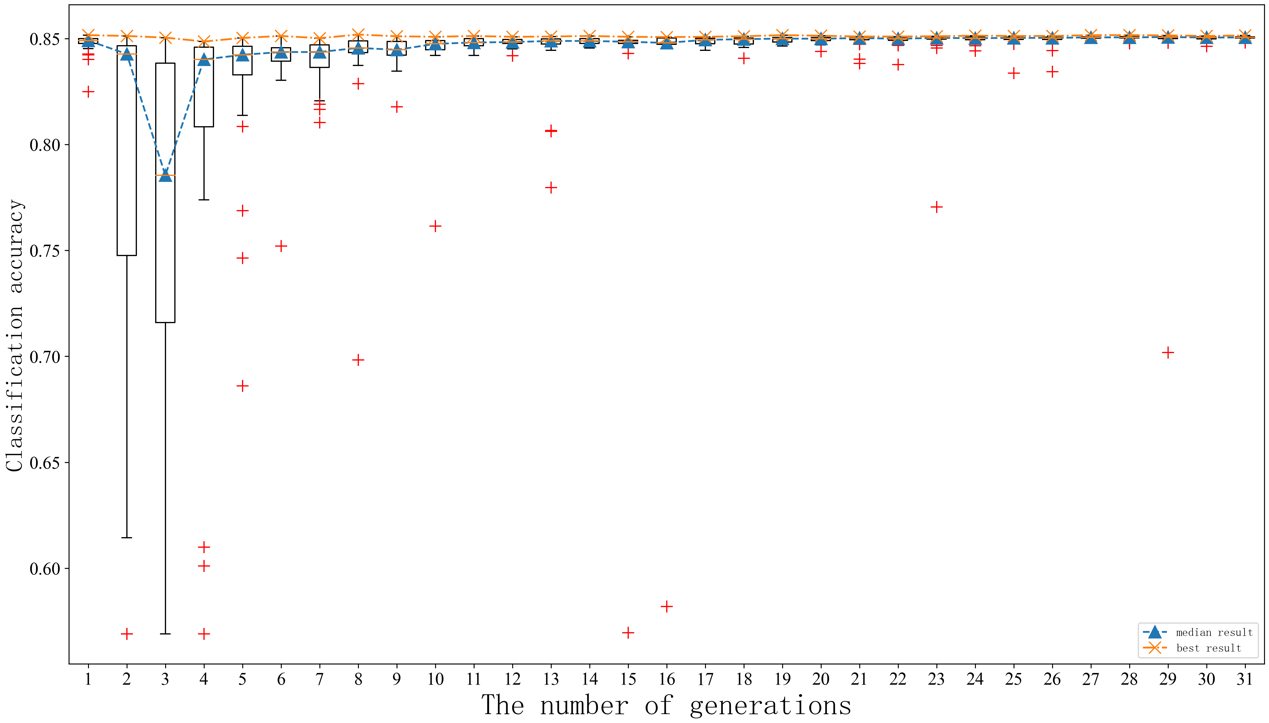


图 9 卷积部分演化结果

表 3 初代与演化后个体对比

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

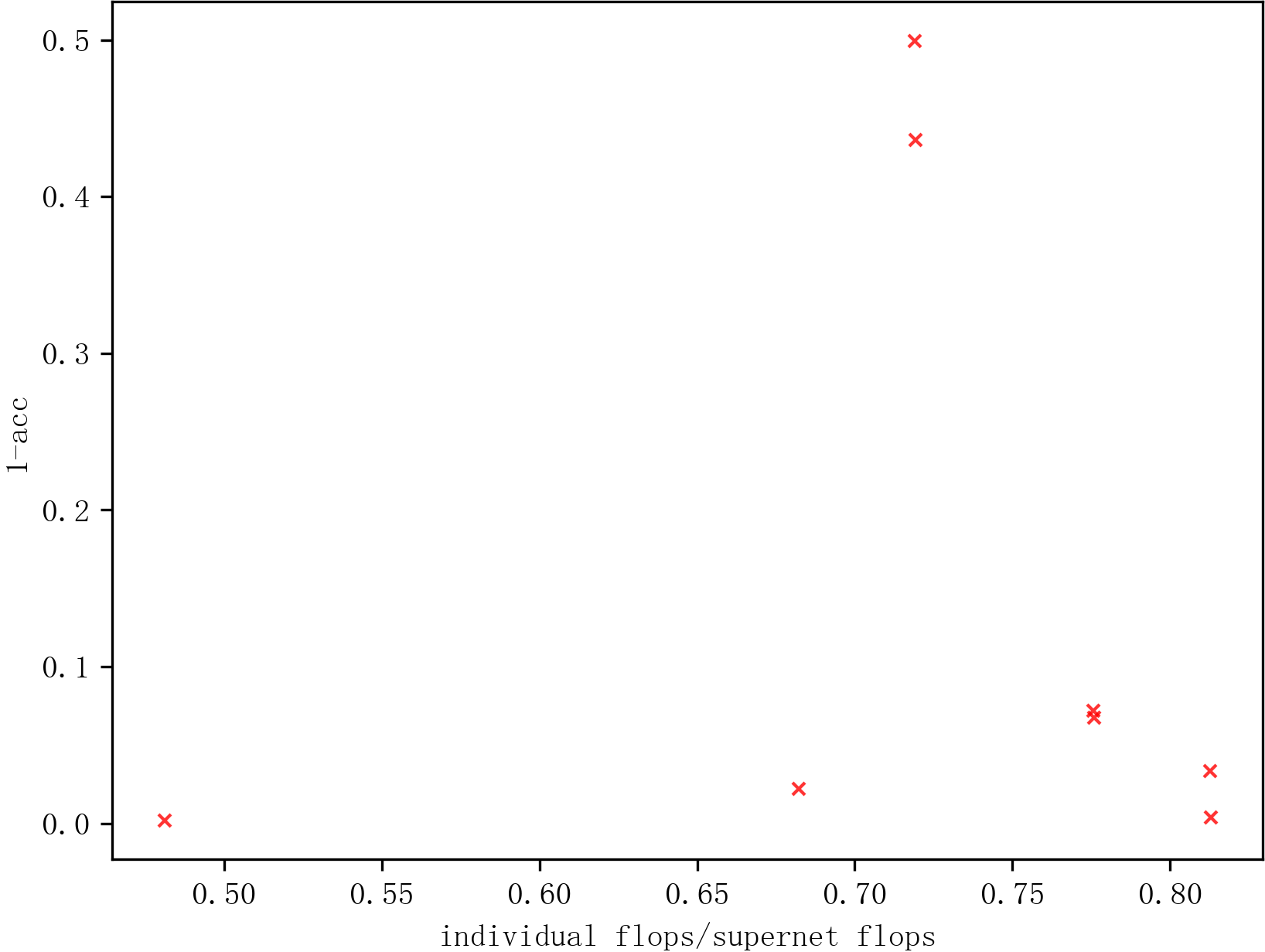


图 10 图卷积部分演化结果

我们对比了是否使用图数据的方向信息对入侵检测的影响。图坍缩分类所需要的池化矩阵是由子图的拉普拉斯矩阵的特征向量组成。我们也对比了是否对拉普拉斯矩阵正则化对入侵检测的影响。从图11可以看到，同时使用图数据方向信息和拉普拉斯的正则化的结果并不好。去掉图数据的方向信息会让入侵检测的效果变得更好，所以更多的使用图数据信息并不一定会提升入侵检测的结果。而去掉拉普拉斯矩阵正则化也会让入侵检测效果变好，而且也会减少数据预处理的计算量。所以，将图数据的方向信息和拉普拉斯矩阵的正则化信息加入搜索空间是合理的。从表1中可以看到这三个网络最终的训练结果。

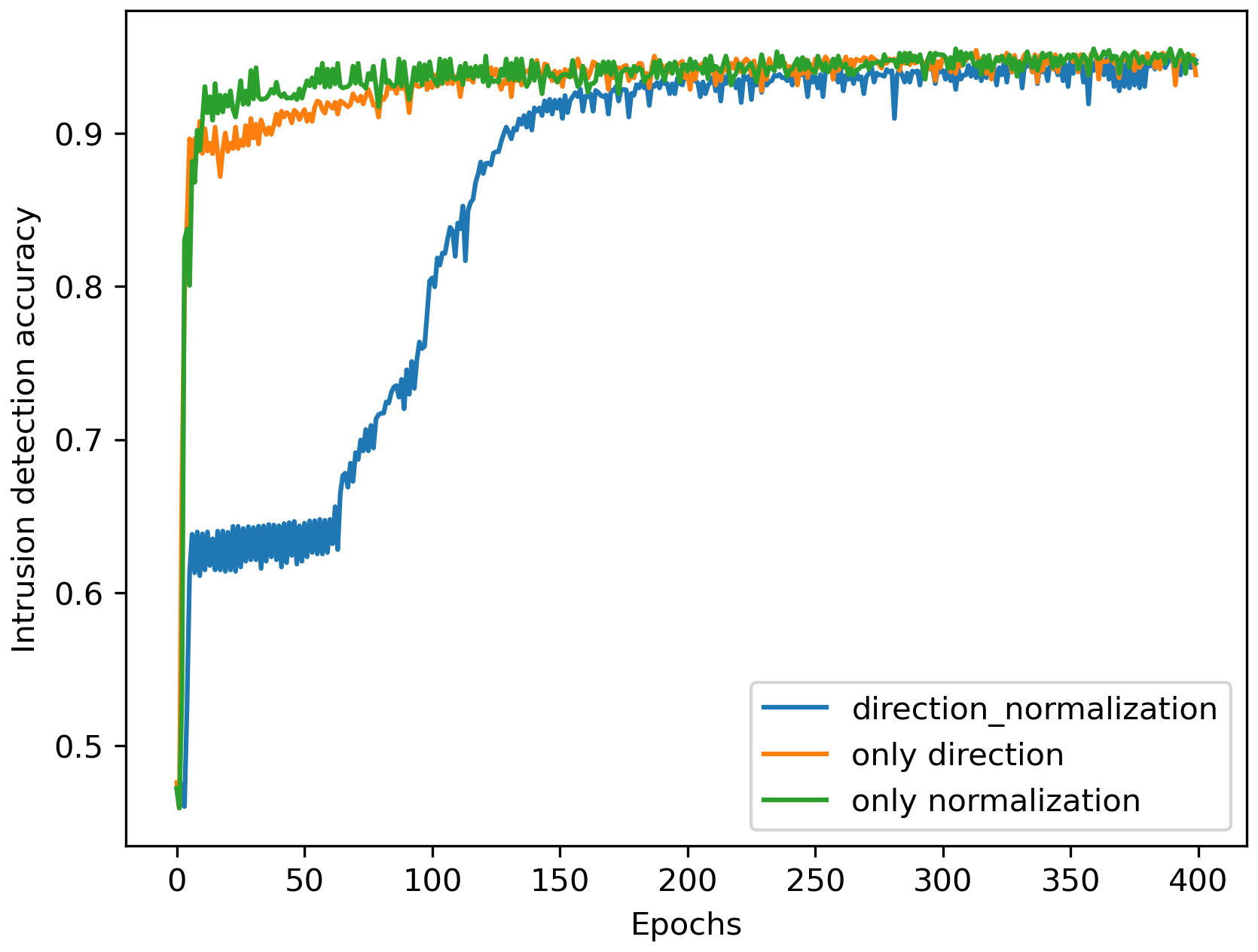


图 11 图数据处理对入侵检测的影响

Table comparison results of three GNN

|  |  |
| --- | --- |
| 基因型 | 入侵检测精度 |
| 1100 0000 000 0021 0124 | 94.86% |
| 1000 0000 000 0021 0124 | 95.43% |
| 0100 0000 000 0021 0124 | 95.52% |

本实验对比实验结果为LSTM和DNN和单独CNN和单独GNN

[[49](#_ENREF_49)]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。网络的损失函数定义为CAN报文相邻两帧对应每个位的交叉熵损失。在网络验证阶段使用取所有位的最大的损失值大小作为是否是入侵报文的依据。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。

[[75](#_ENREF_75)]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。

[[48](#_ENREF_48)]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。[48]中的数据集是作者自己收集的数据，在公共数据集的分类精度并没有达到作者自己实验中的精度，比我们演化出来的卷积网络部分的的精度要低。

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% | 3.6M |
| DNN | 95.36% | 6.8M |
| GNN | 96.43% | 54.3M |
| CNN | 85.06% | 2628.7M |
| 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。本工作同时利用了卷积网络和图网络的优势，虽然得到了非常好的结果，但是模型复杂度偏高，可以考虑图网络和LSTM网络的结合模型。本工作是分别对卷积网络和图网络进行架构搜索，可以考虑同时对强化学习、卷积、和图网络的整体架构进行搜索，以提升检测精度，降低复杂度。增加多目标演化算法的目标函数的数据量，比如False positive rate 和True positive rate。

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] S. Woo, H. J. Jo, and D. H. Lee, "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, no. 2, pp. 993-1006, 2015, doi: 10.1109/TITS.2014.2351612.

[3] M. M. K. Tareq, O. Semiari, M. A. Salehi, and W. Saad, "Ultra Reliable, Low Latency Vehicle-to-Infrastructure Wireless Communications with Edge Computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 9-13 Dec. 2018 2018, pp. 1-7, doi: 10.1109/GLOCOM.2018.8647367.

[4] 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.

[5] 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.

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

[7] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network Representation Learning: A Survey," *IEEE Transactions on Big Data,* vol. 6, no. 1, pp. 3-28, 2020, doi: 10.1109/TBDATA.2018.2850013.

[8] P. Cui, X. Wang, J. Pei, and W. Zhu, "A Survey on Network Embedding," *IEEE Transactions on Knowledge and Data Engineering,* vol. 31, no. 5, pp. 833-852, 2019, doi: 10.1109/TKDE.2018.2849727.

[9] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph Neural Networks in Recommender Systems: A Survey," *ACM Comput. Surv.,* 2022, doi: 10.1145/3535101.

[10] Q. Xie, J. Huang, P. Du, M. Peng, and J.-Y. Nie, "Graph Topic Neural Network for Document Representation," presented at the Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 2021. [Online]. Available: <https://doi.org/10.1145/3442381.3450045>.

[11] M. Koutrouli, E. Karatzas, D. Paez-Espino, and G. A. Pavlopoulos, "A Guide to Conquer the Biological Network Era Using Graph Theory," (in English), *Frontiers in Bioengineering and Biotechnology,* Review vol. 8, 2020-January-31 2020, doi: 10.3389/fbioe.2020.00034.

[12] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open,* vol. 1, pp. 57-81, 2020/01/01/ 2020, doi: <https://doi.org/10.1016/j.aiopen.2021.01.001>.

[13] Q. Li, X. Peng, Y. Qiao, and Q. Peng, "Learning label correlations for multi-label image recognition with graph networks," *Pattern Recognition Letters,* vol. 138, pp. 378-384, 2020/10/01/ 2020, doi: <https://doi.org/10.1016/j.patrec.2020.07.040>.

[14] M. Sun, S. Zhao, C. Gilvary, O. Elemento, J. Zhou, and F. Wang, "Graph convolutional networks for computational drug development and discovery," *Briefings in Bioinformatics,* vol. 21, no. 3, pp. 919-935, 2019, doi: 10.1093/bib/bbz042.

[15] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 33, no. 01, pp. 922-929, 07/17 2019, doi: 10.1609/aaai.v33i01.3301922.

[16] J. Skarding, B. Gabrys, and K. Musial, "Foundations and Modeling of Dynamic Networks Using Dynamic Graph Neural Networks: A Survey," *IEEE Access,* vol. 9, pp. 79143-79168, 2021, doi: 10.1109/ACCESS.2021.3082932.

[17] A. Taheri, K. Gimpel, and T. Berger-Wolf, "Learning to Represent the Evolution of Dynamic Graphs with Recurrent Models," presented at the Companion Proceedings of The 2019 World Wide Web Conference, San Francisco, USA, 2019. [Online]. Available: <https://doi.org/10.1145/3308560.3316581>.

[18] X. Wang *et al.*, "Traffic Flow Prediction via Spatial Temporal Graph Neural Network," in *Proceedings of The Web Conference 2020*: Association for Computing Machinery, 2020, pp. 1082–1092.

[19] 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>.

[20] S. Falkner, A. Klein, and F. Hutter, "BOHB: Robust and Efficient Hyperparameter Optimization at Scale," 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/falkner18a.html>.

[21] Y. Yuan, W. Wang, G. M. Coghill, and W. Pang, "A novel genetic algorithm with hierarchical evaluation strategy for hyperparameter optimisation of graph neural networks," *arXiv preprint arXiv:2101.09300,* 2021.

[22] Y. Jaafra, J. Luc Laurent, A. Deruyver, and M. Saber Naceur, "Reinforcement learning for neural architecture search: A review," *Image and Vision Computing,* vol. 89, pp. 57-66, 2019/09/01/ 2019, doi: <https://doi.org/10.1016/j.imavis.2019.06.005>.

[23] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A Survey on Evolutionary Neural Architecture Search," *IEEE Transactions on Neural Networks and Learning Systems,* pp. 1-21, 2021, doi: 10.1109/TNNLS.2021.3100554.

[24] X. Dong and Y. Yang, "One-shot neural architecture search via self-evaluated template network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3681-3690.

[25] Y. Chen *et al.*, "Renas: Reinforced evolutionary neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4787-4796.

[26] M. Shi *et al.*, "Evolutionary architecture search for graph neural networks," *arXiv preprint arXiv:2009.10199,* 2020.

[27] Y. Gao, H. Yang, P. Zhang, C. Zhou, and Y. Hu, "Graph Neural Architecture Search," in *IJCAI*, 2020, vol. 20, pp. 1403-1409.

[28] K. Zhou, Q. Song, X. Huang, and X. Hu, "Auto-gnn: Neural architecture search of graph neural networks," *arXiv preprint arXiv:1909.03184,* 2019.

[29] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research,* vol. 20, no. 1, pp. 1997-2017, 2019.

[30] Y. Gao, H. Yang, P. Zhang, C. Zhou, and Y. Hu, "Graphnas: Graph neural architecture search with reinforcement learning," *arXiv preprint arXiv:1904.09981,* 2019.

[31] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review,* vol. 53, no. 8, pp. 5455-5516, 2020/12/01 2020, doi: 10.1007/s10462-020-09825-6.

[32] S. Mirjalili, "Evolutionary algorithms and neural networks," *Studies in computational intelligence,* vol. 780, 2019.

[33] A. Bhandare and D. Kaur, "Designing Convolutional Neural Network Architecture Using Genetic Algorithms," *International Journal of Advanced Network, Monitoring and Controls,* vol. 6, no. 3, pp. 26-35, 2021, doi: doi:10.21307/ijanmc-2021-024.

[34] X. Wang, Y. Jin, and K. Hao, "Evolving Local Plasticity Rules for Synergistic Learning in Echo State Networks," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 31, no. 4, pp. 1363-1374, 2020, doi: 10.1109/TNNLS.2019.2919903.

[35] Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang, "Surrogate-Assisted Evolutionary Deep Learning Using an End-to-End Random Forest-Based Performance Predictor," *IEEE Transactions on Evolutionary Computation,* vol. 24, no. 2, pp. 350-364, 2020, doi: 10.1109/TEVC.2019.2924461.

[36] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Completely Automated CNN Architecture Design Based on Blocks," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 31, no. 4, pp. 1242-1254, 2020, doi: 10.1109/TNNLS.2019.2919608.

[37] Q. Ye, Y. Sun, J. Zhang, and J. Lv, "A Distributed Framework for EA-Based NAS," *IEEE Transactions on Parallel and Distributed Systems,* vol. 32, no. 7, pp. 1753-1764, 2021, doi: 10.1109/TPDS.2020.3046774.

[38] Y. Sun, G. G. Yen, and Z. Yi, "Improved Regularity Model-Based EDA for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation,* vol. 22, no. 5, pp. 662-678, 2018, doi: 10.1109/TEVC.2018.2794319.

[39] J. Chen, J. Gao, Y. Chen, M. B. Oloulade, T. Lyu, and Z. Li, "GraphPAS: Parallel Architecture Search for Graph Neural Networks," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*: Association for Computing Machinery, 2021, pp. 2182–2186.

[40] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *arXiv preprint arXiv:1811.12560,* 2018.

[41] Q. He and X. Hou, "WD3: Taming the Estimation Bias in Deep Reinforcement Learning," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 9-11 Nov. 2020 2020, pp. 391-398, doi: 10.1109/ICTAI50040.2020.00068.

[42] J. Pan, X. Wang, Y. Cheng, and Q. Yu, "Multisource Transfer Double DQN Based on Actor Learning," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 29, no. 6, pp. 2227-2238, 2018, doi: 10.1109/TNNLS.2018.2806087.

[43] 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>.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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>.

[49] 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.

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

[51] 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>.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] Z. Yin, W. Gross, and B. H. Meyer, "Probabilistic Sequential Multi-Objective Optimization of Convolutional Neural Networks," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 9-13 March 2020 2020, pp. 1055-1060, doi: 10.23919/DATE48585.2020.9116535.

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

[58] 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>.

[59] 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.

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

[61] 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>.

[62] Z. Lu, K. Deb, E. Goodman, W. Banzhaf, and V. N. Boddeti, "NSGANetV2: Evolutionary Multi-objective Surrogate-Assisted Neural Architecture Search," Cham, 2020: Springer International Publishing, in Computer Vision – ECCV 2020, pp. 35-51.

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

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

[65] 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.

[66] 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.

[67] 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.

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

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

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.