Computer Science and Application 计算机科学与应用, 2025, 15(2), 102-109
Published Online February 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.152037

Hans汉斯

基于智能合约漏洞检测的元模型算法

龚　伟

江西理工大学信息工程学院，江西　赣州

收稿日期：2025年1月11日；录用日期：2025年2月12日；发布日期：2025年2月24日

摘　要

本文提出了一种基于元学习的智能合约漏洞检测方法，通过结合多层感知机模型和MetaSGD优化器，实现了高效的训练和检测性能。该方法能够在少量样本和有限梯度更新的条件下，快速适应新任务，并取得良好的泛化效果。同时，神经网络策略的引入进一步加速了梯度强化学习的微调过程。实验结果表明，该方法在智能合约漏洞检测任务中具有较强的实用性和可靠性，为智能合约安全提供了一种高效的解决方案。

关键词

智能合约漏洞，元学习，MetaSGD

Metamodel Algorithm Based on Smart Contract Vulnerability Detection

Wei Gong

School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou Jiangxi

Received: Jan. 11th, 2025; accepted: Feb. 12th, 2025; published: Feb. 24th, 2025

Abstract

This paper proposes a meta-learning-based smart contract vulnerability detection method, which achieves efficient training and detection performance by combining a multi-layer perceptron model and a MetaSGD optimizer. This method can quickly adapt to new tasks and achieve good generalization effects under the conditions of a small number of samples and limited gradient updates. At the same time, the introduction of the neural network strategy further accelerates the fine-tuning process of gradient reinforcement learning. Experimental results show that this method has strong practicality and reliability in smart contract vulnerability detection tasks, and provides an efficient solution for smart contract security.

文章引用: 龚伟. 基于智能合约漏洞检测的元模型算法[J]. 计算机科学与应用, 2025, 15(2): 102-109.
DOI: 10.12677/csa.2025.152037

Keywords

Smart Contract Vulnerability, Meta-Learning, MetaSGD

1. 引言

自 2008 年中本聪发布比特币白皮书以来，区块链作为一种分布式账本技术被广泛应用，现如今已经走向“区块链 3.0”时代，并写入国务院发布的《“十四五”数字经济发展规划》中。智能合约作为区块链上部署的去中心化应用程序，帮助区块链实现安全高效的信息存储、资产管理和价值转移。近年来，针对区块链智能合约的安全威胁频出，不仅造成了巨大的经济损失，也影响了区块链的信用体系[1]。

如今区块链智能合约已经被广泛应用于各种场景，如互联网金融、供应链平台、质量溯源等。与此同时，针对区块链智能合约的威胁种类也日益增多。随着新型威胁种类的出现，准确的未知威胁检测通常需要安全专家进行手动审计或模拟攻击场景。在这种情况下，可用的未知威胁样本少之又少。基于传统机器学习、深度学习的威胁检测方法如 ContractWard、Peculiar、AFS 等，具有较高的可扩展性以及较好的检测效率等优势，然而这些方法都需要足够数量的精确且高质量的样本与标签，所以传统机器学习、深度学习方法无法充分训练未知威胁检测模型以达到预期的检测效果[2]。本文从两部分介绍了本次研究的内容。首先，在数据提取方面，通过编写脚本从智能合约源码中提取有价值的特征，为模型训练提供高质量的数据集，并构建了一个专门的数据库。其次，在模型构建与训练上，采用多层感知机(MLP)结构，并引入归一化层来提升训练效果，同时通过 Dropout 层随机关闭部分神经元，以降低过拟合风险并增强模型的泛化能力。为进一步优化模型性能，本文创新性地引入了自学习优化算法 MetaSGD，通过动态调整学习率，使得模型能够在不同任务和数据集上自适应优化，从而提高未知威胁检测的精度与效率。

2. 相关工作

2.1. 元模型

元学习(Meta-learning)，也被称为“学习如何学习”，与传统的深度学习模型有所不同。传统的深度学习目标是学习一个数学模型，用于特定任务的预测，而元学习则关注于优化学习过程本身，即“如何更快、更好地学习一个模型”。在元学习中，我们不仅关注预测结果，而是通过学习如何调整学习策略，从而提高模型在不同任务上的学习效率。

在本次研究中，我们采用了经典的模型无关元学习(MAML)框架进行训练。MAML 的核心思想是，不论任务的具体类型如何，它都可以通过优化初始模型参数，使得模型在面对新任务时能够迅速适应并取得良好的表现。该方法适用于多种学习场景，包括分类、回归甚至强化学习(RL)。MAML 通过调整模型的初始参数，使得模型能够在少量数据的情况下更好地适应新任务，而无需重新训练整个模型结构[3]。

与传统的机器学习方法不同，元学习的输入是一个个任务(task)，而非单一的数据点。这意味着，模型的训练并不是针对某一特定任务，而是通过多任务学习来捕捉如何快速适应不同任务的能力。例如，人类通过多次经历不同的任务(如猫狗分类、苹果香蕉分类等)，逐渐形成区分物体的能力。类似地，MAML 通过训练多个任务来优化模型，使得它能够快速适应新的、未见过的任务。这种方法能够有效提升模型

在少样本情况下的学习能力，进而提高模型的泛化能力。

2.2. 元模型优化算法

元学习模型(Model-Agnostic Meta-Learning, MAML)是一种通用的元学习算法，它的核心思想是通过在多个任务上进行训练，使得模型能够通过少量的数据快速适应新任务。在 MAML 中，优化方法(或优化器)扮演了至关重要的角色，帮助模型在不同任务之间快速调整并优化参数。以下是一些常用的优化方法，这些方法可以与 MAML 算法结合使用，以便提高在新任务上的学习效果：

2.2.1. 梯度下降法

梯度下降法是最基础的优化方法之一，广泛应用于机器学习和深度学习中。在 MAML 中，梯度下降法通过计算损失函数相对于模型参数的梯度来更新参数，以最小化损失函数。传统的梯度下降法通常是批量梯度下降(Batch Gradient Descent)，它每次计算完整的数据集上的梯度，并用来更新参数。然而，这种方法在大规模数据集上计算开销较大，因此，通常会采用小批量梯度下降(Mini-batch Gradient Descent)来平衡计算效率和收敛速度。

在 MAML 框架中，梯度下降的更新步骤通常是在每个任务的训练过程中进行多次参数更新，以使得模型的参数在不同任务之间能够更好地迁移和适应。

2.2.2. 随机梯度下降法

随机梯度下降(SGD)是梯度下降法的一种变体，它通过在每个训练样本或小批量(mini-batch)上计算梯度并更新模型参数。与批量梯度下降不同，SGD 每次更新仅基于一个样本或小批量样本的梯度，这使得 SGD 更加高效，尤其是在处理大规模数据时。MetaSGD (Meta-learning Adaptive Stochastic Gradient Descent)是对传统 SGD 的扩展，特别设计用于元学习任务。它结合了动态学习率调整机制，使得优化过程更加高效。在 MAML 框架下，MATESGD 不仅能够在每个任务上使用 SGD 进行基本的梯度更新，还通过自适应调整每个参数的学习率来应对任务间的差异。通过在训练过程中对学习率的动态调整，MATESGD 能够帮助模型更快速地适应新任务，提升了快速学习和迁移能力，尤其在面对复杂或稀疏的数据时，能有效避免过拟合并加速收敛。

2.2.3. 自适应优化算法

自适应优化算法是一类根据梯度的历史动态调整学习率的优化方法，这些算法可以有效地应对梯度消失或梯度爆炸等问题，提高学习效率。常见的自适应优化算法包括 Adam、Adagrad、RMSprop 等，这些算法通过调整每个参数的学习率，使得优化过程更加平稳。

- Adam (Adaptive Moment Estimation)是自适应优化算法中最常用的一种。它结合了动量法和 RMSprop 的思想，使用参数的梯度一阶矩(动量)和二阶矩(方差)来计算每个参数的自适应学习率。Adam 在训练过程中能够根据每个参数的历史梯度调整学习率，从而提高了收敛速度并减少了调参的复杂性。
- Adagrad 通过为每个参数动态调整学习率来有效处理稀疏数据，它会将学习率与梯度的历史累积进行平衡，从而确保在参数更新时不会过快地调整。
- RMSprop 则改进了 Adagrad 的缺点，通过引入一个指数衰减的平方梯度平均值，避免了 Adagrad 在训练后期学习率过低的问题。

3. 模型设计

本次设计的元模型开发流程涵盖了数据处理、模型定义、参数优化以及元训练与测试等多个阶段如图 1 所示。首先，通过对原始数据进行特征提取和标准化处理，生成适用于模型训练与测试的数据集。

在模型构建上，采用多层感知机(MLP)模型，包含五层全连接层，每层配备批量归一化(Batch Normalization)以稳定训练过程，并结合 ReLU 激活函数提升非线性表达能力，同时通过 Dropout 机制防止过拟合以增强泛化能力。在优化阶段，使用 MetaSGD 算法对任务级子网络进行动态学习率调整，使模型能够针对不同任务学习参数初始化与个性化学习率，提升自适应性。最后，在元训练阶段，通过多任务迭代优化主网络的初始参数和学习率，在元测试阶段则将模型应用于新任务并验证其性能。整个流程自数据到测试层层递进，确保了模型在多任务学习中的高效性与可扩展性[3]。

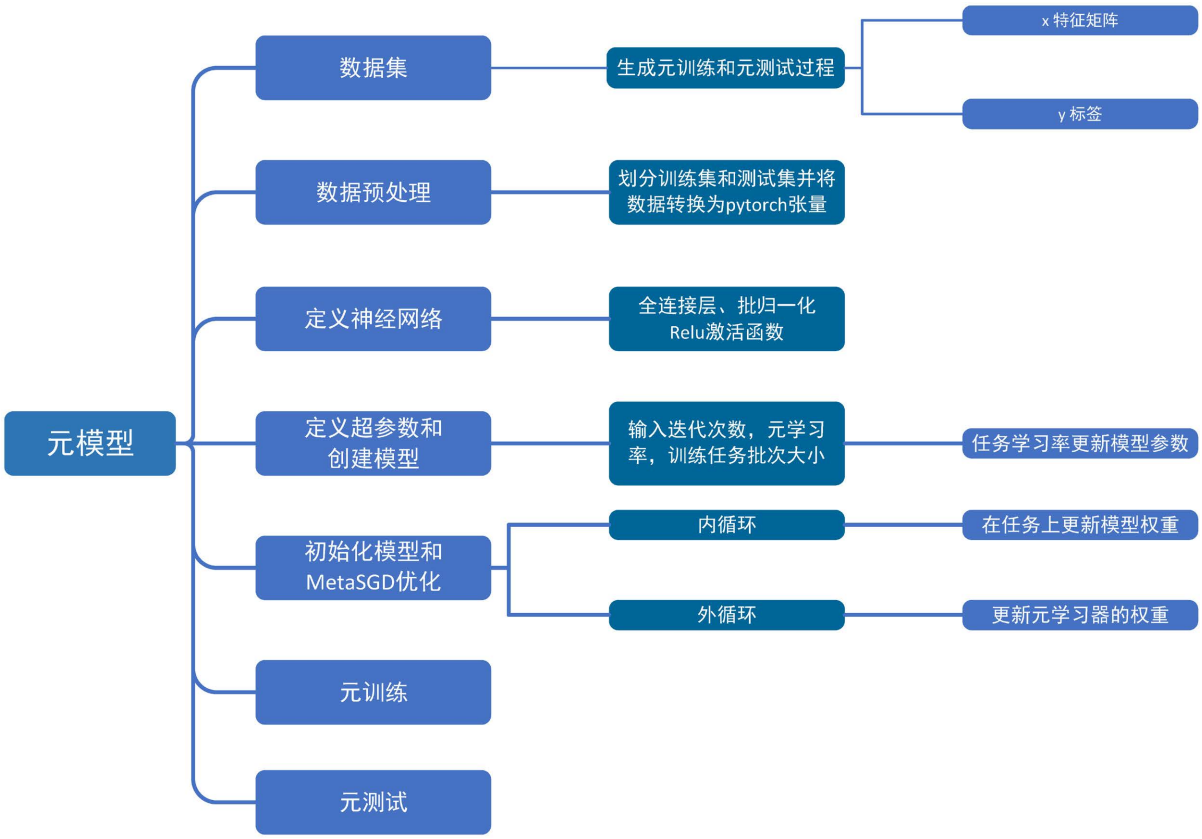

Figure 1. Metamodel structure diagram
图 1. 元模型结构图

在本次实验中，我们也定义了前向传播(forward)方法，逐步实现输入数据在神经网络中的处理流程。具体而言，输入数据依次通过全连接层、批量归一化层、ReLU 激活函数和 Dropout 层进行逐层转换，这一过程会重复多次，最终在输出层得到预测结果。在此基础上，为了进一步提升模型的性能，我们保持基础神经网络模型结构不变，并针对多种类型的元模型及其相应优化器进行了测试。同时，我们为每个任务构建了子神经网络，并在初始化阶段为每个子网络设置参数。在每次迭代中，这些子网络的参数会根据主网络的状态进行同步初始化，从而实现主网络与子网络的协同优化。特别地，在子网络完成梯度计算后，通过梯度增强策略对其进行二次梯度更新，这不仅优化了子网络的学习过程，还为主网络的元更新提供了更多的反馈信息。这种设计有效地增强了主网络的全局优化能力，从而提升了整个模型在多任务场景中的适应性和表现力。

在本次设计的 MetaSGD 模型中，输入包括多层感知机(MLP)模型和一个预先定义的学习率。在 adapt 方法中，模型根据损失函数对每个参数进行更新，计算梯度后调整模型参数。如果 θ_i 为 True，那么每个

参数都有独立的可学习学习率，否则所有参数共享一个学习率。通过这种方式，MetaSGD 可以根据任务的不同要求，动态地调整每个参数的学习率，以实现更灵活和高效的优化过程。

MetaSGD 的核心思想是学习每个参数的个性化学习率。对于每个模型参数 θ_i，我们引入一个学习率 α_i，并通过以下公式(1)更新每个参数。

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \alpha_i^{(t)} \nabla_{\theta_i} \mathcal{L}\left(\theta_i^{(t)}\right) \tag{1}$$

其中，$\theta_i^{(t)}$ 是第 i 个参数在第 t 次迭代时的值，$\alpha_i^{(t)}$ 是第 i 个参数在第 t 次迭代时的学习率，$\nabla_{\theta_i} \mathcal{L}\left(\theta_i^{(t)}\right)$ 是基于当前参数 $\theta_i^{(t)}$ 计算的梯度，$\mathcal{L}(\theta)$ 是损失函数。在该公式中，MetaSGD 通过动态调整每个参数的学习率 $\alpha_i^{(t)}$ 来使模型参数的更新更加精确，从而提高训练的效率和性能。

为了使每个参数的学习率能够自适应调整，MetaSGD 采用了一种基于梯度的学习率更新机制。

$$\alpha_i^{(t+1)} = \alpha_i^{(t)} - \eta \nabla_{\alpha_i} \mathcal{L}\left(\alpha_i^{(t)}\right) \tag{2}$$

学习率的更新公式如下公式(2)所示，其中 $\alpha_i^{(t)}$ 是第 i 个参数在第 t 次迭代时的学习率，η 是学习率更新的步长，控制学习率更新的速率，$\nabla_{\alpha_i} \mathcal{L}\left(\alpha_i^{(t)}\right)$ 是基于当前学习率 $\alpha_i^{(t)}$ 的梯度。通过这个公式，MetaSGD 可以在训练过程中不断调整每个参数的学习率 α_i，使其在不同的任务和数据集上自动优化，帮助模型更快地收敛，提升性能。

MetaSGD 的核心优势在于其动态学习率机制，即为每个参数引入独立的学习率，并通过梯度下降优化这些学习率，从而实现对不同参数的个性化优化。相比于固定学习率的方法，MetaSGD 的动态学习率能够根据不同任务需求自适应地调整参数的更新幅度，从而加速模型的收敛过程并提升训练效率。例如，对于梯度变化剧烈的参数，MetaSGD 可以适当降低学习率 α_i 以避免参数震荡，而对于收敛较慢的参数，则可以通过增加学习率加速优化过程。这种动态调整的能力在多任务学习中尤为重要，不仅能够减少训练中的参数干扰，还能在不同任务之间实现更好的平衡[4]。此外，动态学习率还能缓解优化过程中的梯度消失和梯度爆炸问题，确保模型在训练后期仍能有效更新，从而提升整体性能。

在智能合约漏洞检测任务中，不同类型的漏洞具有显著的特征分布差异，例如重入漏洞依赖于调用栈深度和跨函数调用的上下文，而整数溢出漏洞则更多依赖于数据流的精确分析。MetaSGD 的动态学习率机制能够针对这些差异化特征，为每个参数分配最优的学习率，从而在特征提取过程中对重要信息进行重点学习[5]。例如，在检测时间戳依赖漏洞时，MetaSGD 可通过调整学习率适应时间相关特征的局部波动，从而增强模型对时间依赖性的理解[6]。此外，在多任务检测场景中，MetaSGD 的个性化优化策略能够显著减少不同漏洞类型之间的特征干扰，确保每个任务都能获得最优的模型适应性。因此，MetaSGD 不仅提升了模型对已知漏洞的检测精度，还增强了其在处理未知威胁时的泛化能力，使其在智能合约漏洞检测的复杂场景中展现出卓越的性能。通过这些机制，MetaSGD 有效填补了传统固定学习率方法在复杂任务下的不足，为智能合约的安全分析提供了一种更高效的解决方案。

4. 实验数据集

在本次实验中，数据集来源于 SmartBugs-Wild，包含 5000 个智能合约，涵盖了多种漏洞类型，包括 1000 个整数溢出漏洞、1000 个时间戳依赖漏洞、1000 个重入漏洞、1000 个访问控制漏洞以及 1000 个其他类型的漏洞(如授权错误和逻辑漏洞等)。为了进行元学习任务的训练与测试，我们将数据集划分为多个任务，其中每个漏洞类型(如整数溢出、时间戳、重入、访问控制和其他类型)作为一个独立的任务。训练阶段中，模型将接触到所有五种漏洞类型的任务，每个任务的支持集包含 300 个合约(占每类任务的 30%)，查询集包含 300 个合约(占每类任务的 30%)，以帮助模型学习从不同漏洞中提取特征并适应多任务检测

场景。测试阶段使用剩余的 40%数据集进行评估，其中包括所有四种漏洞类型(整数溢出、时间戳依赖、重入和访问控制)的样本。该阶段重点验证模型在整数溢出漏洞任务上的检测性能，同时评估其对其他漏洞类型的泛化能力，以全面分析模型在多任务环境下的综合表现。通过利用 SmartBugs-Wild 的丰富样本与多样化漏洞种类，本实验设计旨在显著提升模型的检测性能和泛化能力。

同时，我们构建了合约的控制流图(CFG)和数据流图(DFG)，如图 2 所示，这些图结构帮助提取最有效的数据特征，以增强漏洞检测的精度。最后，所有提取的特征被转换为 one-hot 编码形式，并统一调整为等长的输入数据，便于输入到模型进行训练与测试。在此基础上，模型将通过对重入漏洞、整数溢出漏洞及其他漏洞类型的训练和测试[7]，进一步评估其在不同漏洞类型上的检测能力，尤其是在整数溢出漏洞上的专门测试，旨在验证模型对未知威胁的识别与适应能力[8]。

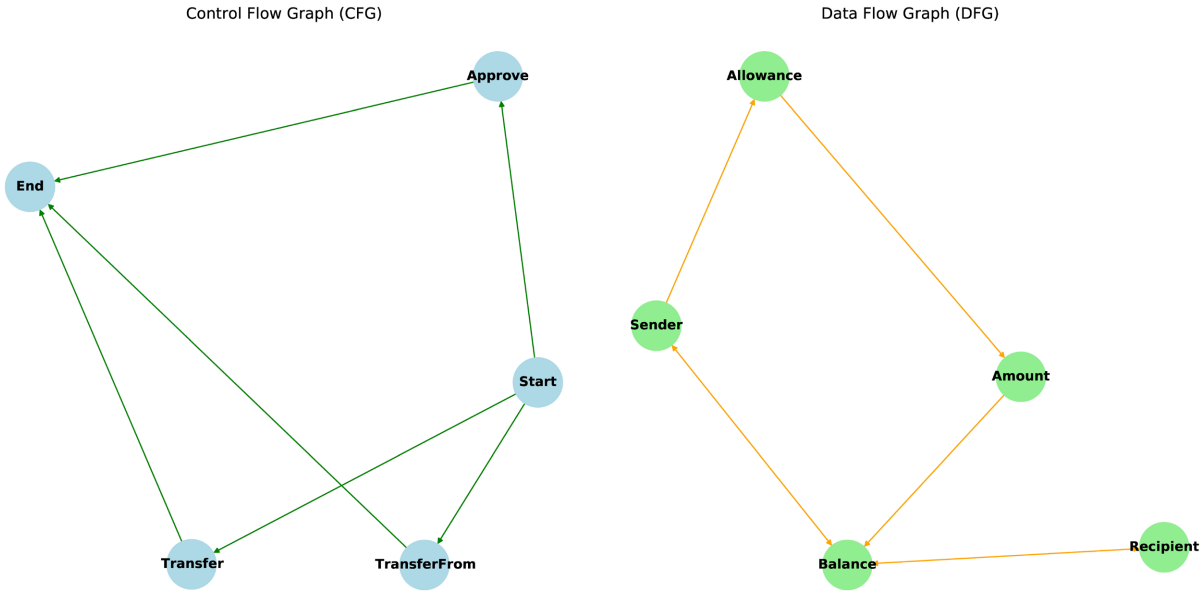

Figure 2. Control flow graph and data flow graph
图 2. 控制流图和数据流图

5. 实验结论和分析

在本次实验中，我们分别对数据集中包含的四种漏洞类型(整数溢出、时间戳依赖、重入和访问控制)进行了全面测试。实验结果表明，该模型在所有任务上的表现均优异，其中检测率在整数溢出漏洞任务中达到最高的 90%，而其他漏洞类型的检测率均保持在 85%至 89%之间，充分体现了模型在多任务环境下的高效性和稳定性。以整数溢出漏洞检测任务为例，模型的准确率为 88%，表明能够正确预测 88%的样本；平均精准度为 87%，说明在所有预测为漏洞的样本中，87%的合约确实存在漏洞；平均召回率为 88%，反映了模型能够有效识别绝大多数实际存在漏洞的样本；平均 F1 分数为 86%，综合了精准度和召回率，表明模型在整体检测效果上的优越性能。

为进一步验证模型的有效性，我们将其与其他主流方法进行了对比。首先，基于图神经网络(GNN) [9]的漏洞检测方法在相同任务中的检测率平均为 81%，召回率为 79%，F1 分数为 80%，整体表现略逊于我们的方法。经过充分分析，我们认为 GNN 的核心优势在于利用图结构建模智能合约中的数据流和依赖关系。然而，GNN 的性能很大程度上依赖于输入图的质量和图结构的复杂性。在智能合约漏洞检测任务中，一些漏洞(如重入漏洞和整数溢出漏洞)涉及复杂的数据流和跨函数调用的依赖关系，这些复杂关系可能导

致 GNN 的图表示能力不足，进而限制其对特征的全面捕获。其次，与 FSL (基于少样本学习的检测方法) [10]相比，该方法通过元学习框架和注意力机制在小样本任务中具有较好的泛化能力，其在整数溢出漏洞任务中的检测率为 83%，召回率为 82%，F1 分数为 82%，尽管其在少样本场景中具有优势，但在多任务检测和未知漏洞适应性上仍不及我们提出的方法。相比之下，我们的方法结合了 MetaSGD 动态学习率和双循环元学习策略，在不同任务中能够动态调整学习过程，并通过优化的特征提取机制对复杂的数据流关系进行更有效的建模与捕获。具体而言，MetaSGD 的动态学习率机制为不同参数赋予了个性化优化能力，而双循环元学习策略显著增强了模型对多任务场景的适应性。正因如此，我们的方法在处理复杂依赖关系和特征多样性时展现出更强的鲁棒性和更高的泛化能力，从而在检测性能上展现出卓越的性能表现。

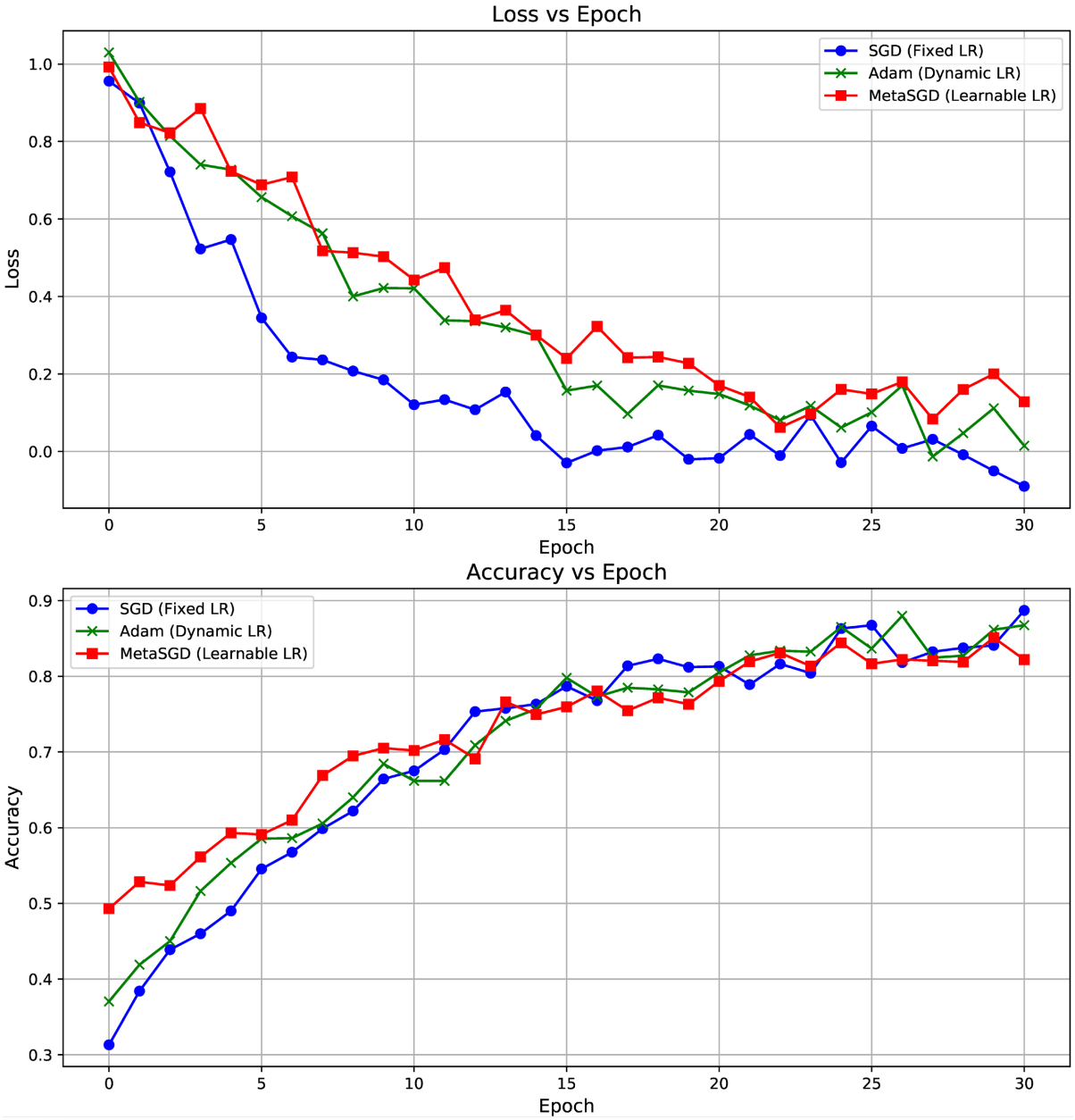

Figure 3. Comparison of MetaSGD and traditional optimization methods in terms of loss and accuracy
图 3. MetaSGD 与传统优化方法在损失与准确率上的表现对比图

在本次实验中，我们还对比了三种不同优化器(SGD, Adam, MetaSGD)在固定学习率和动态学习率策略下的性能表现，以分析它们对模型训练的影响。通过记录训练损失(Loss)和准确率(Accuracy)随训练轮次(Epoch)的变化，我们能够清晰地观察到各优化器的收敛速度和最终性能表现差异。MetaSGD 以其动态可学习的学习率调整机制，在多个参数维度上实现了更精确的优化，有效提升了模型的训练效率和适应能力。

从图 3 可以看出，在损失曲线中，MetaSGD 的损失下降速度明显快于其他优化器，表明其能够快速找到较优解，并在后期保持较稳定的下降趋势。而 SGD (固定学习率)的损失下降最慢，且收敛较不稳定，表明固定学习率可能限制了模型的优化效率。Adam 优化器表现介于两者之间，通过动态调整学习率实现了一定程度的收敛加速。在准确率曲线中，MetaSGD 的准确率提升最快，且最终准确率最高，表现出优异的泛化能力和任务适应性。相比之下，SGD 的准确率增长缓慢且波动较大，说明其在固定学习率下难以有效适应复杂任务。Adam 则在准确率提升和稳定性上优于 SGD，但仍次于 MetaSGD。这表明，MetaSGD 的可学习率机制显著提升了模型在未知任务上的优化效率和检测能力。综上所述，实验验证了该元模型在整数溢出漏洞检测中的有效性和适应性，同时也进一步证明了 MetaSGD 优化器和多层感知机模型在新任务中的优异表现，为模型的性能提升提供了重要支持。

6. 总结

本文围绕区块链智能合约威胁检测问题，提出了一种结合多层感知机(MLP)和 MetaSGD 优化器的模型方法，以解决传统方法对样本和标签依赖较高的问题。通过多层感知机的归一化层提升训练稳定性和 Dropout 层降低过拟合风险，增强了模型的泛化能力；同时，MetaSGD 优化器通过动态调整每个参数的学习率，使模型具备自适应优化能力，在少样本和新任务场景中展现了显著优势。在实验中，模型在新漏洞类型——整数溢出漏洞检测中表现出良好的适应性和检测能力，其准确率、精准度、召回率和 F1 分数均达到较高水平，充分证明了其在新任务场景中的优异性能。实验结果验证了本文方法在快速收敛、任务扩展性及鲁棒性方面的有效性，为解决区块链智能合约未知威胁检测提供了新的思路和工具。未来工作将进一步扩展至更多漏洞类型检测，并结合特征提取技术优化模型，为区块链安全提供更强的技术支持。

参考文献

[1] 丁诗琪, 陈正奎, 黄海. 基于数据流图和混合网络模型的智能合约漏洞检测[J]. 软件工程, 2025, 28(1): 52-56.

[2] 苏盛锋, 光焱, 郭旺, 等. 智能合约漏洞检测分析研究综述[J]. 信息工程大学学报, 2024, 25(5): 586-592.

[3] 杨浩杰, 鲁强. MKML: 用于零样本常识问答的多知识元学习算法[J/OL]. 计算机工程与应用, 2025: 1-16. http://kns.cnki.net/kcms/detail/11.2127.TP.20250108.1822.002.html, 2025-01-21.

[4] 李思颖. 基于长短期记忆网络和元学习方法的蜜罐合约检测研究[D]: [硕士学位论文]. 长沙: 湖南大学, 2023.

[5] 黄志强. 智能合约的安全漏洞检测方法研究[D]: [硕士学位论文]. 杭州: 杭州电子科技大学, 2024.

[6] 王顺. 智能合约漏洞检测技术研究[D]: [硕士学位论文]. 杭州: 杭州电子科技大学, 2023.

[7] 涂良琼, 孙小兵, 张佳乐, 等. 智能合约漏洞检测工具研究综述[J]. 计算机科学, 2021, 48(11): 79-88.

[8] 倪远东, 张超, 殷婷婷. 智能合约安全漏洞研究综述[J]. 信息安全学报, 2020, 5(3): 78-99.

[9] 哈焱. 基于图神经网络(GNN)的漏洞检测算法及应用研究[J]. 喀什大学学报, 2024, 45(6): 68-71.

[10] 何道敬, 丁柯. 智能合约未知威胁的检测方法[J]. IEEE 物联网期刊, 2024, 11(3): 4430-4441.