小 型 微 型 计 算 机 系 统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2025-0121
2026 年 4 月 第 4 期
Vol.47 No.4 2026

一种基于多模态时序数据融合的工业物联网系统异常检测方法

郑泽梁[1],吕明琪[2],陈铁明[2],朱添田[1],王 飞[3]

[1](浙江工业大学 计算机科学与技术学院,杭州 310023)
[2](浙江工业大学 地理信息学院,浙江 德清 313200)
[3](中国石油大学(华东) 控制科学与工程学院,山东 青岛 266000)
E-mail:zeliangzheng@foxmail.com

摘 要: 工业物联网(IIoT)系统异常检测通过实时监控工业设备和网络的状态,及时发现潜在问题,是保障生产连续性和设备安全,实现从传统的定期维护到预测性维护的转变的重要手段.由于工业物联网系统的异常往往难以大量捕获和标注,因此无监督学习是目前的主流方法.然而,由于无监督学习技术的限制,现有方法大多仅考虑单一模态的数据(如多维时序传感数据),导致对复杂隐蔽的异常的检测能力不足.为此,本文提出了一种基于多模态时序数据融合的工业物联网异常检测方法.该方法考虑工业物联网系统中常见的两类异构时序数据(传感器数据与网络流量数据),首先构建独立的传感数据自编码器与网络流量数据自编码器,以捕获各自的时序特征.随后,通过一种面向无监督学习的集成框架对不同自编码器的重构误差进行建模,并利用全局自编码器融合多模态信息,以实现跨模态异常检测.实验结果表明,相较于传统的单一模态检测方法,本文方法在 ToN_IoT 数据集上的 F1 分数提升了 7% ~40%,有效提高了异常检测的准确性和鲁棒性.

关 键 词: 异常检测;工业物联网;自动编码器;多模态时序数据融合
中图分类号: TP391 **文献标识码:** A **文 章 编 号:** 1000-1220(2026)04-0990-09

Anomaly Detection Method for Industrial Internet of Things Systems Based on Multimodal Time-series Data Fusion

ZHENG Zeliang[1], LÜ Mingqi[2], CHEN Tieming[2], ZHU Tiantian[1], WANG Fei[3]
[1](College of Computer Science, Zhejiang University of Technology, Hangzhou 310023, China)
[2](College of Geographic Information, Zhejiang University of Technology, Deqing 313200, China)
[3](College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266000, China)

Abstract: Industrial Internet of Things (IIoT) anomaly detection, which involves real-time monitoring of the status of industrial equipment and networks to identify potential issues in a timely manner, is an important means of ensuring production continuity and equipment safety. It also enables the transition from traditional periodic maintenance to predictive maintenance. However, since anomalies in IIoT systems are often difficult to capture and label in large quantities, unsupervised learning has become the mainstream approach. Nevertheless, due to the limitations of unsupervised learning techniques, most existing methods only consider single-modal data (such as multidimensional time-series sensor data), resulting in insufficient detection capabilities for complex and hidden anomalies. To address this issue, this study proposes an IIoT anomaly detection method based on the fusion of multimodal time-series data. The method considers two common types of heterogeneous time-series data in IIoT systems: sensor data and network traffic data. First, independent autoencoders for sensor data and network traffic data are constructed to capture their respective temporal features. Subsequently, a fusion framework for unsupervised learning is employed to model the reconstruction errors from different autoencoders. A global autoencoder is then used to integrate multimodal information to achieve cross-modal anomaly detection. Experimental results show that compared with traditional single-modal detection methods, the proposed method achieves a 7% to 40% increase in F1 score on the ToN_IoT dataset, effectively improving the accuracy and robustness of anomaly detection.

Keywords: anomaly detection; industrial internet of things; autoencoder; multimodal time-series data fusion

0 引 言

近年来,工业物联网(IIoT)的出现彻底改变了传统工业格局.通过将先进的连接性和智能技术融入制造过程,IIoT 系统整合了大量相互连接的设备和传感器,共同实现对工业操作的高效监控与控制.这种范式转变不仅开启了生产力的新纪元,还在异常检测领域带来了新的挑战.异常检测是指识别与预期行为的偏离,这种偏离可能意味着故障、攻击或其他意

收稿日期:2025-03-14 收修改稿日期:2025-04-17 基金项目:国家自然科学基金项目(62372410)资助;杭州市重点科研计划项目(2024SZD0220)资助;浙江省尖兵计划项目(2024C01066,2025C01013)资助;嘉兴市科技计划项目(2025CGZ046)资助. 作者简介:**郑泽梁**,男,2000 年生,硕士,CCF 会员,研究方向为工业物联网异常检测;**吕明琪**(通信作者),男,1982 年生,博士,教授,CCF 会员,研究方向为数据驱动安全、时空数据挖掘;**陈铁明**,男,1978 年生,博士,教授,CCF 会员,研究方向为网络空间安全;**朱添田**,男,1992 年生,博士,副教授,CCF 会员,研究方向为网络攻击检测;**王 飞**,男,1988 年生,博士研究生,工程师,CCF 会员,研究方向为工业互联网.

外事件[1]. 检测异常能够实现从传统的定期维护到预测性维护的转变,防止设备故障、提高流程效率并提升整体性能.

为实现人、设备和系统之间的信息交互,IIoT 系统通过网络连接大量监控设备和传感器,并收集能够反映生产过程和状态的数据. 由于工业流程和 IIoT 数据的复杂性高,理解异常状态困难,且异常事件本身十分罕见,因此在海量的 IIoT 数据中找出并标注异常状态成本十分高昂. 在实践中,从真实的 IIoT 系统中获取标记异常样本通常非常困难甚至不可能,因此大多数研究集中在无监督学习技术上[2,3].

在传统机器学习技术中,现有方法通常分为两个步骤:首先从工业数据中提取大量特征,然后使用离群点检测算法(例如聚类[4,5]、单类支持向量机[6]、孤立森林[7]等)发现异常. 然而,特征工程依赖于领域知识,IIoT 数据的高维性和动态性使得设计有效特征变得极为困难.

近年来,随着深度学习在自动从原始数据中学习特征方面的能力不断增强,其在 IIoT 异常检测任务中的应用也越来越广泛. 大多数研究集中在基于编码器-解码器框架[8,9]的无监督深度学习模型上. 具体而言,这些模型使用编码器学习多变量时间序列数据的低维语义特征,并使用解码器重构正常数据,然后通过重构误差检测异常. 此外,由于物联网传感数据具有时间性,大多数现有研究基于循环神经网络(RNN)[8-10]设计编码器和解码器. 还有少数研究尝试从 IIoT 网络流量数据中检测异常,具体方法是从原始网络数据包[11]中提取统计和时间特征,并基于自编码器构建异常检测模型. 这些特征通过解码器重构,并利用重构误差进行网络流量异常检测[12,13]. 多层感知机(MLP)[13]和长短期记忆网络(LSTM)[14]等模型被用于实现自编码器,以适应这些统计和时间特征. 然后,基于解码器对这些特征进行重构,并利用重构误差进行网络流量异常检测.

尽管在使用无监督深度自编码器进行 IIoT 异常检测方面取得了进展,但现有方法大多仅考虑单一模态的时序数据,缺少同时考虑传感数据和网络流量数据进行 IIoT 系统异常检测的研究,限制了对可能跨越两个领域的异常的全面理解和检测. 现代 IIoT 系统通常表现出相互关联的行为,一个领域的异常可能会在整个基础设施中产生连锁反应,因此需要一个能够有效整合两种数据源的统一框架. 然而,这是一项具有挑战性的任务,原因如下:

首先,传感器数据和网络流量数据属于两种截然不同的时序数据模态,其在数据结构、采样方式、时间依赖性及关联模式等方面存在显著差异. 传感器数据通常是等间隔采样的多变量时间序列,其特征维度相对固定,且不同传感器之间通常存在物理或逻辑上的相关性,例如工业设备的温度、压力、振动等传感器数据往往相互影响、相互作用. 而网络流量数据则是从不同时刻的网络通信事件中捕获而来,其数据包的到达时间并不均匀,且数据维度随协议类型、流量特征等因素动态变化. 例如,TCP 流量包含序列号、窗口大小等特征,而 UDP 流量缺乏这些信息. 此外,网络流量数据还包含离散的类别特征(如源 IP 地址、目的端口号)以及统计特征(如流量速率、数据包大小分布),与传感器数据的连续特征形式差异较大. 这些固有的不一致性导致两种数据难以直接对齐到同一时间尺度,也无法直接映射到统一的特征空间,使得传统的单一时序数据建模方法难以适用.

其次,尽管已有研究广泛探讨了异构数据的融合方法,但它们主要集中在有监督学习任务上. 有监督的异构数据融合通常依赖于明确的目标函数,例如特征级融合通过将传感数据和网络流量数据的特征向量进行拼接或加权组合,形成一个新的特征表示;又或者是决策级融合,它通过对来自不同数据源的分类结果进行投票或加权平均来做出最终决策. 然而,对于无监督学习任务,这些有监督的异构数据融合方式往往不可行. 主要原因在于无监督学习缺乏明确的标签信息,无法直接利用标签来指导数据融合过程. 例如,在特征级融合中,由于没有标签信息,很难确定如何有效地拼接或加权不同数据源的特征;在决策级融合中,由于没有分类结果,无法进行投票或加权平均;同时,由于传感数据和网络流量数据具有不同的模态,无法使用同一个自动编码器重构这两种完全不兼容的时序数据.

为此,本文提出了一种基于多模态时序数据融合的工业物联网系统异常检测方法,该方法通过融合物联网传感数据和网络流量数据来克服上述挑战. 首先,该方法使用独立的自编码器来捕获传感数据与网络流量数据各自的时序特征. 然后,通过一种面向无监督学习的集成框架跨不同数据模态融合物联网传感数据和网络流量数据,以实现异常检测.

1　相关工作

1.1　工业异常检测

工业数据是典型的多变量时间序列数据,具有数据量大、维度高和动态性强的特点,因此传统的单变量异常检测方法[2-6]无法有效工作.

为了适应复杂的多变量时间序列数据,大多数现有研究采用基于学习的技术(包括机器学习和深度学习),这些技术大致可以分为两类,即监督学习技术和无监督学习技术. 在监督学习技术方面,Griffin 等人[15]提出了一种基于神经网络和决策树的异常检测方法,用于检测多个处理过程中的异常. Nanduri 等人[10]提出使用循环神经网络来检测可能降低飞行安全因素的异常事件. Janssens 等人[16]提出了一种基于卷积神经网络(CNN)的特征学习系统,用于检测旋转机械的故障状态.

尽管监督学习技术可以通过明确学习异常样本中的潜在模式来更好地识别异常,但由于缺乏标记的异常训练数据,其在实践中难以实施. 由于正常工业数据可以轻松大规模获取,因此无监督学习技术大多被应用于工业多变量时间序列异常检测任务. 在早期阶段,传统的无监督机器学习方法如聚类、单类支持向量机(One-Class SVM)和孤立森林(iForest)被应用. 例如,Amruthnath 和 Gupta[17]应用多种聚类算法进行异常检测(例如 K-Means、模糊 C 均值). 这些算法的异常检测可以定义为识别偏离标准行为的过程. Diez-Olivan 等人[18]提出了一种基于单类支持向量机的异常检测方法,通过获取样本与分离超平面之间的距离来计算异常分数,从而检测传感器数据中的异常. Joshi 等人[19]基于隐马尔可夫模型(HMM)进行异常检测,通过提取特征并计算模型生成的状态序列中的异常概率来构建分类器. Li 等人[20]提出了一种基于单类支持

向量机的异常检测方法,通过在自监督深度表示上构建生成式单类分类器来检测图像中的异常.

然而,传统机器学习技术的性能严重依赖于特征工程,但由于工业数据的高维度和高动态性,设计有效的特征非常困难.为了解决这一问题,近年来深度学习技术被广泛应用于工业异常检测,其中生成式编码器-解码器深度神经网络是最常采用的策略.例如,Kingma 等人[21]应用变分自编码器(VAE)通过重建数据并分析重建数据与源数据之间的残差来检测异常.Lu 等人[22]使用堆叠自编码器检测旋转机械部件中的异常.Zhang 等人[23]提出了 MSCRED,该方法使用 ConvLSTM 自编码器来学习由多尺度签名矩阵在不同时间步长中所表征的系统运行模式的多个层次.Yin 等人[24]将卷积神经网络(CNN)和循环自编码器集成在一起,用于检测物联网系统中的异常.具体来说,他们采用两阶段滑动窗口策略设计编码器,以实现更好的特征提取.Muneer 等人[25]提出了一种基于深度自编码器神经网络(DANN)的混合模型,该模型包含 5 层,用于检测真实世界燃气轮机数据集中的异常.尽管编码器-解码器深度神经网络在工业异常检测中取得了有希望的结果,但如果能够明确学习工业数据与网络流量数据之间的关系,并在两个数据源之间适当平衡,仍有改进的空间.

1.2 跨域数据融合

物联网和网络流量领域的数据由多种模态组成,每种模态都有不同的表示形式、分布、规模和密度.例如,工业传感数据通常由生产线上多个传感器设备收集的信息构成,其中可能包括设备温度和压力等数据.另一方面,网络流量数据来自数据包分析,包含数据传输的源地址和目的地址等详细信息.因此,跨模态的数据融合成为了一项挑战.数据融合方法可以分为基于阶段的融合方法、基于特征级别的融合方法以及基于语义含义的融合方法[26].

基于阶段的方法在数据融合任务的不同阶段使用不同的数据集.因此,这些数据集之间的联系较为松散,并且不存在确保其跨模态一致性的先决条件.例如,Xiao 等人[27]利用空间轨迹来检测停留点,并基于周围的兴趣点(POI)将其转换为特征向量.这些特征向量通过分层聚类形成树形结构,能够表示用户的位置历史记录,并便于基于用户的层次图来测量用户之间的相似度.

至于基于特征级别的方法,最先进的方法是基于深度神经网络(DNN)从不同的数据集中学习统一的特征表示.例如,Ngiam 等人[28]引入了一种深度自编码器架构,用于捕捉不同模态之间的中间特征表示,探索了 3 种学习设置:跨模态学习、共享表示学习和多模态融合,这有效地改进了单模态表示,并捕捉了多种模态之间的相关性.Nagrani 等人[29]引入了基于注意力瓶颈的融合(FSN)作为一种优化基于自注意力的多模态交互的策略,通过有限数量的瓶颈潜在变量来传递信息,从而增强早期拼接,进而提高融合性能并降低计算成本.

基于特征的数据融合方法将特征视为数值或分类值,而不考虑其语义含义,而基于语义含义的方法则理解每个数据集的内在含义,识别特征之间的关系,并通过融入对每个数据集的重要性及其相互作用的理解来提供可解释且有意义的融合.例如,Zheng 等人[30]提出了一种基于协同训练的模型,用于对城市范围内的空气质量进行细粒度推断,该模型利用了由空气质量、气象信息、交通数据、兴趣点(POI)和道路网络这五个数据集.尽管在跨域数据融合的研究方面已经取得了重大进展,但目前缺乏针对工业物联网(IIoT)中使用无监督学习任务进行多模态融合的研究.

2 方 法

2.1 方法概述

在本章中,本文首先定义相关概念和问题,然后介绍本文方法的架构.

定义 1. (传感数据样本):一个时间步 t 的传感样本表示为 $X_t \in \mathbb{R}^{N\times W}$,其中 N 是传感器的数量,W 是采样时间长度.并且 X_t 是 N 个传感器在时间区间 $(t-W,t]$ 内的多维时间序列的一个子序列.

定义 2. (网络流量数据样本):一个时间步 t 的网络流量样本表示为 $Y_t \in \mathbb{R}^{K\times W}$,其中 K 是网络流量的特征维度,W 是时间窗口的长度.Y_t 是在时间区间 $(t-W,t]$ 内捕获的多个时间窗口的网络流量序列.每个时间窗口的网络流量数据包包含从网络数据包中提取的 K 个统计特征.本文将 Y_t 中的每个单变量时间序列视为一个统计特征.

定义 3. (IIoT 数据样本):一个时间步 t 的 IIoT 数据样本记为 $S_t=(X_t,Y_t)$,它由同一时间周期内的一个传感样本和一个网络流量样本组成.

图 1 给出了一个 IIoT 数据样本的实例.可以看出,IIoT 数据样本包含了两种模态的数据:时序数据(对应传感数据)和半结构化文本数据(对应网络流量数据).该实例中,传感数据是一个 5 维时序数据,而网络流量数据包含了源 IP 地址、目的 IP 地址、协议等关键字段.

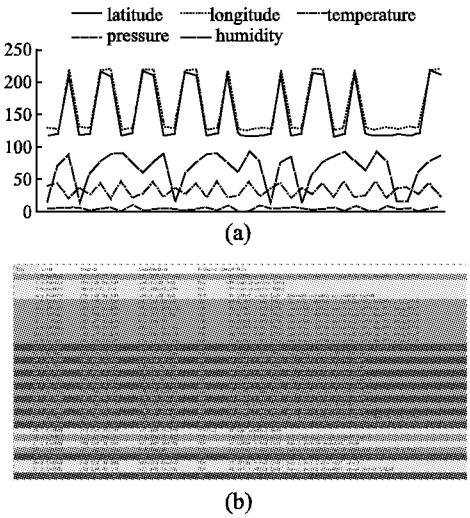

图 1 传感与网络流量数据案例图

Fig. 1 Sensing and network traffic data case diagram

问题定义:本文中的工业物联网系统异常检测定义为从 IIoT 数据样本中学习,以获得一个函数 f,它以 S_t 作为输入,并产生一个输出值,表示在时间步 t 内是否存在异常.

图 2 展示了本文方法的架构,它包括 3 个模块,即传感自编码器(IoT-AE)模块、网络自编码器(Network-AE)模块和融合模块.传感自编码器模块对传感样本集应用自助抽样

(bootstrap sampling),创建多个训练子集,然后基于这些子集训练多个传感数据自编码器. 每个传感数据自编码器包含 3 层,即图卷积(GCN)层、编码器层和解码器层. 网络自编码器模块也对网络流量样本集应用自助抽样,创建多个训练子集,然后基于这些子集训练多个网络流量数据自编码器. 每个网络流量数据自编码器包含 3 层,即特征提取层、编码器层和解码器层. 融合模块可以被视为传感自编码器模块和网络自编码器模块之上的一个元学习器(meta-learner). 它以多个自编码器输出的重建均方根误差(RMSE)作为输入,并训练一个全局自编码器来重建这些 RMSE 值.

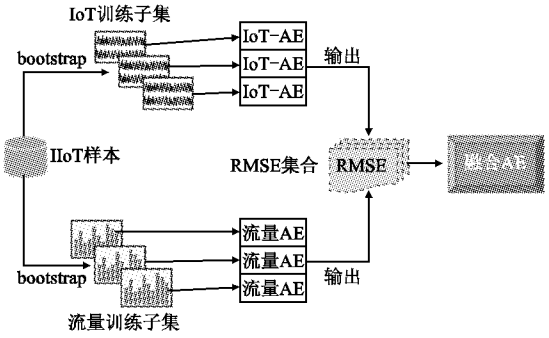

图 2　本文方法的架构图

Fig. 2　Architecture of the method in this study

在实时异常检测阶段,给定一个 IIoT 数据样本 $S_t=(X_t, Y_t)$,X_t 和 Y_t 将分别输入到传感自编码器模块和网络自编码器模块的多个自编码器中,这些自编码器将输出一系列重建 RMSE 值. 这些 RMSE 值随后将输入到融合模块中,并输出一个全局 RMSE 值,用于确定是否存在异常.

2.2　主干自编码器

本文方法本质上是一个自编码器的集成框架,在融合之前,分别为传感数据和网络流量数据构建了自编码器. 由于这两种数据模态的特征差异,本文为它们使用了不同的骨干自编码器.

2.2.1　基于传感数据的自编码器

传感数据的骨干自编码器由 3 层组成,即 GCN 层、编码器层和解码器层. GCN 层用于学习不同传感器之间的潜在关联;编码器层用于学习传感数据的时空模式;解码器层则用于重构输入数据.

1)GCN 层

传感数据通常呈现为高维时序数据. 首先,这些代表不同传感器的维度之间存在潜在的关联和相互影响. 例如,一些异常检测任务需要同时考虑多个传感器的数据才能有效识别异常. 其次,不同传感器的数据对异常检测任务的影响程度也可能不同. 例如,在液压系统中,压力数据通常比其他传感器的数据对异常检测更为重要. 针对这些问题,GCN 层首先利用图结构来捕捉传感数据不同维度之间的关联(步骤 1),然后通过 GCN 子网络学习这些关联的强度(步骤 2).

步骤 1. (传感图构建):传感器图表示为 $G=(V,E,F)$,其中每个节点代表传感数据的一个维度(即一个传感器),每条边 $e_{i,j}\in E$ 表示传感器 v_i 和 v_j 之间的关联,F 表示特征集,每个元素 f_i 代表节点 v_i 的原始特征(即来自传感器 v_i 的单一时序数据). 为了构建传感器图,是否存在关联可以通过领域知识来决定. 然而,由于领域知识往往难以获取,若没有领域知识,本文设计 G 为一个完全连接的图.

步骤 2. (关联强度学习):本文应用 GCN 学习 G 中每对节点之间的关联强度. GCN 通过聚合每个节点的邻居信息来更新节点的表示. 具体而言,对于每个节点 v_i,其更新后的嵌入向量 g_i 通过式(1)计算. 其中 $N(i)$ 表示节点的邻居集合,d_i 和 d_j 分别表示节点 v_i 和 v_j 的度,W 是可学习的权重矩阵,σ 是非线性激活函数.

通过这两个步骤,每个传感样本 X_t 将被表示为一个特征矩阵 $GX_t\in\boldsymbol{R}^{N\times W}$,其中 GX_t 的第 i 行代表节点 v_i 的嵌入向量(即 g_i).

$$g_i=\sigma\left(\sum_{j\in N(i)}\frac{1}{\sqrt{d_i d_j}}Wg_j\right) \tag{1}$$

2)编码层

自编码器通过编码器将原始的传感样本映射到一个低维的潜在特征空间,然后通过解码器将潜在特征重构回原始样本空间. 自编码器的训练是通过逐步减小原始样本与重构样本之间的误差来实现的.

由于潜在特征的维度低于原始样本,潜在特征可以视为原始样本的主要模式. 在异常检测任务中,自编码器是基于正常样本(或大部分为正常样本)进行训练的,因此训练后的自编码器的潜在特征可以表示正常样本的主要模式. 如果一个重构样本与原始样本有很大的偏差,说明原始样本不符合正常样本的主要模式,可能表示异常.

考虑到传感样本的时间特性,本文使用 LSTM 作为编码器. 如图 3 所示,给定传感样本 $X_t=[x_{t-W},x_{t-W+1},\cdots,x_t]$($x_t\in\mathbb{R}^{N\times 1}$ 为时间步 t 的传感器读数),本文首先将 X_t 输入到 GCN 层,并获得更新后的特征矩阵 $GX_t=[gx_{t-W},gx_{t-W+1},\cdots,gx_t]$. 然后,将 GX_t 输入到 LSTM 子网络,并根据式(2)计算出 W 个隐藏状态向量 $h^{ie}_{t-W},h^{ie}_{t-W+1},\cdots,h^{ie}_t$,最终将最后一个隐藏状态向量 h^{ie}_t 作为编码器层的输出.

$$h^{ie}_t=\mathrm{LSTM}(h^{ie}_{t-1},gx_t) \tag{2}$$

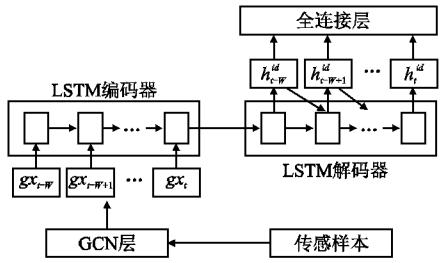

图 3　基于 LSTM 的自编码器架构

Fig. 3　Architecture of the LSTM autoencoder

3)解码层

如图 3 所示,解码器将 h^{ie}_t 作为输入,并使用 LSTM 子网络根据式(3)输出 W 个隐藏状态向量 $h^{id}_{t-W},h^{id}_{t-W+1},\cdots,h^{id}_t$. 之后,使用一个全连接子网络将 $h^{id}_{t-W},h^{id}_{t-W+1},\cdots,h^{id}_t$ 重构为与原始样本 $\tilde{X}_t\in\boldsymbol{R}^{N\times W}$ 相同的维度.

$$h^{id}_t=\mathrm{LSTM}(h^{id}_{t-1},h^{ie}_t) \tag{3}$$

为了训练自编码器,本文将损失函数定义为原始输入样

本 X_t 与重构样本 $\tilde{X}_t$ 之间的误差,如式(4)所示:

$$Loss_I = \sqrt{\frac{\sum_{i=0}^{N}\sum_{j=0}^{W}(X_t[i,j] - \tilde{X}_t[i,j])^2}{NW}} \tag{4}$$

2.2.2 基于网络流量的自编码器

针对网络流量数据的自编码器由 3 个主要部分组成:特征提取层、编码器层和解码器层. 这种结构设计旨在高效地处理网络流量数据的复杂性和多样性,同时提取出对异常检测有用的特征.

1)特征提取层

特征提取层是网络流量数据自编码器的第 1 步,其目标是从原始网络数据包中提取有意义的统计特征. 这一过程包括 3 个关键步骤:样本生成、特征提取和特征选择.

样本生成:通过工具(如 tcpdump 和 wireshark)捕获网络数据包,并选择在特定时间窗口 $(t-W,t]$ 内具有相同源 IP、目的 IP、源端口、目的端口和协议的网络数据包,生成网络流量样本. 这一过程确保了样本的代表性,能够反映网络流量的行为模式.

特征提取:本文使用 CICFlowMeter 工具从每个网络流量样本中提取统计特征. 这些特征包括时间持续性、正向/反向传输的总数据包数量/字节数、平均数据包长度、每秒传输的数据包数量/字节数、连续数据包时间间隔的均值/标准差等. 这些统计特征能够从多个角度描述网络流量的特性,为后续的异常检测提供丰富的信息.

特征选择:特征选择是机器学习模型中的一项重要功能,其目的是筛选出对异常检测最有价值的特征. 为了保证特征选择结果的稳定性,本文采用了一种简单的集成策略来计算每个特征的重要性得分. 如式(5)所示,给定 n 种机器学习算法,对于每个特征 f_k,本文首先计算每种算法的分类准确率,记为 acc_i,并使用每种算法计算该特征的重要性得分,记为 $score_i(k)$. 然后,将这些得分加权平均,得到特征 f_k 的最终重要性得分,记为 $I(k)$. 通过这种方式,本文能够筛选出对异常检测最有价值的前 K 个特征,从而将网络流量样本表示为一个 K 维向量. 特征选择不仅减少了数据的维度,还提高了模型的效率和鲁棒性. 经过特征选择后,本文得到了网络流量样本的前 K 个重要特征. 随后,时间区间 $(t-W,t]$ 内的网络流量样本 $Y_t=[y_{t-W},y_{t-W+1},\cdots,y_t]$.

$$I(k) = \frac{1}{n}\sum_{i=1}^{n} acc_i \times score_i(k) \tag{5}$$

2)编码器层

本文使用 LSTM 作为网络流量样本的编码器. 对于一个网络流量样本 $Y_t=[y_{t-W},y_{t-W+1},\cdots,y_t]$ ($y_t \in \boldsymbol{R}^{K\times 1}$ 为时间步 t 的网络流量统计特征),编码器层的任务是将其映射到一个低维特征空间. 具体而言,LSTM 通过其时间步长的递归结构捕捉网络流量数据中的事件依赖性,并根据公式(6)计算出 W 个隐状态向量,最终将隐状态向量 h_t^{ne} 作为编码器层的输出.

$$h_t^{ne} = \mathrm{LSTM}(y_t, h_{t-1}^{ne}) \tag{6}$$

3)解码器层

解码器层的任务是将 h_t^{ne} 重构为与原始样本相同维度的向量. 本文同样使用 LSTM 作为解码器. 解码器的隐藏状态更新公式如式(7)所示:

$$h_t^{de} = \mathrm{LSTM}(\tilde{y}_t, h_{t-1}^{de}) \tag{7}$$

其中,$\tilde{y}_t$ 是时间步 t 的重构特征向量,h_t^{de} 是时间步 t 的隐藏特征向量,h_{t-1}^{de} 是前一个时间步的隐藏状态向量. 最终,解码器的输出为重构的网络流量样本 $\tilde{Y}_t \in \mathbb{R}^{K\times W}$.

为了训练自编码器,本文定义损失函数为原始输入样本 Y_t 和重构样本 $\tilde{Y}_t$ 之间的误差,如公式(8)所示:

$$Loss_N = \sqrt{\frac{\sum_{i=0}^{N}\sum_{j=0}^{W}(Y_t[i,j] - \tilde{Y}_t[i,j])^2}{NW}} \tag{8}$$

2.3 基于自编码器集成的数据融合

2.3.1 无监督数据融合框架

由于传感数据和网络流量数据通常具有不同的维度和特征,它们无法通过单一的统一自编码器进行处理. 相反,本文使用具有不同结构的自编码器来分别对它们进行建模. 另一方面,从同一个 IIoT 系统中收集的传感数据和网络流量数据通常具有语义相关性. 例如,在智能家居环境中,攻击者可以远程控制门锁. 由于攻击行为涉及入侵家庭智能家居网络并在边缘设备中潜伏,因此线索可能会同时反映在网络流量数据和设备日志数据中. 因此,本文需要一种能够学习传感数据和网络流量数据间语义相关性的数据融合机制.

然而,现有的数据融合框架大多是为监督学习任务设计的. 例如,最流行的几种传统数据融合框架(包括 Bagging[31]、Boosting[32]和 Stacking[33])通过利用分类一致性、分类残差或分类概率分布来进行数据融合,而在无监督学习条件下,目前没有这样的信息可供利用. 深度学习模型通常通过在统一的学习目标中整合不同的子网络来实现数据融合. 不幸的是,在统一的自编码器中同时重建异构数据模态是不可行的.

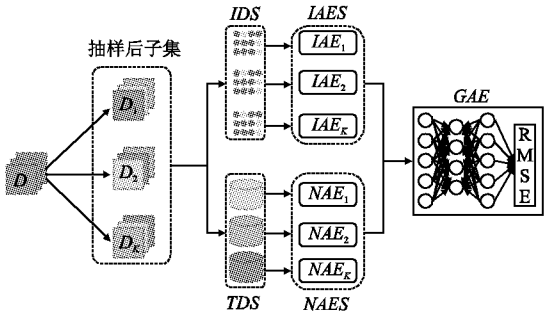

图 4 自编码器的数据融合框架

Fig. 4 Data fusion framework for autoencoders

为了解决这一挑战,本文提出了一种针对不同数据模态自编码器的数据融合框架. 如图 4 所示,该框架由 3 部分组成,即数据采样、局部自编码器训练和全局自编码器训练.

1)数据采样

多个子模型的集成是数据融合的有效方法. 根据以往的研究,这些子模型的多样性是集成有效性的关键条件. 因此,本文采用自助抽样(bootstrap sampling)的方法,分别为传感数据和网络流量数据创建多个训练子集. 具体而言,给定一个训练数据集 D,本文进行 K 次自助抽样迭代. 在每次迭代中,通过从 D 中随机选择 N 个样本(允许重复选择)生成一个训练子集,这意味着数据集 D 中的样本可能在训练子集中被多次选择,也可能一次也不被选中. 通过应用自助抽样,本文可

以为每种数据模态生成多样化的训练子集.

最终,本文可以分别获得传感数据的 K 个训练子集,记为 $IDS=\{ID_1,ID_2,\cdots,ID_K\}$ 和网络流量数据的 K 个训练子集,记为 $TDS=\{TD_1,TD_2,\cdots,TD_K\}$.

2)局部自编码器训练

在获得两种数据模态的训练子集集合 IDS 和 TDS 之后,本文通过以下步骤训练多个自编码器. 首先,对于 IDS 中的每个训练子集 ID_k,本文训练一个用于传感数据的骨干自编码器(记作 IAE_k). 其次,对于 TDS 中的每个训练子集 TD_k,本文训练一个用于网络流量数据的骨干自编码器(记作 NAE_k). 最终,本文得到了一组基于传感数据的局部自编码器 $IAES=\{IAE_1,IAE_2,\cdots,IAE_K\}$ 和一组基于网络流量数据的局部自编码器 $NAES=\{NAE_1,NAE_2,\cdots,NAE_K\}$. 通过这些步骤,本文确保每个训练子集都用于训练一个专门的局部自编码器,并且最终得到的局部自编码器集合能够捕捉 IIoT 数据中的多样化模式.

3)全局自编码器训练

为了融合来自 $IAES$ 和 $NAES$,的多个局部自编码器,本文尝试训练一个全局自编码器来捕捉它们之间的语义相关性和行为模式. 核心思想借鉴了堆叠(Stacking)的方法,即全局自编码器基于局部自编码器的输出进行训练,但采用无监督的方式. 具体步骤如下.

均方根误差(RMSE)生成:给定一个 IIoT 数据样本 $S_t=(X_t,Y_t)$,本文将传感样本 X_t 输入到 $IAES$ 中的每个局部自编码器中. 对于每个局部自编码器 IAE_k,它将输出一个重构样本 $\tilde{X}_t$,并且本文计算 X_t 与重构样本 $\tilde{X}_t$ 之间的均方根误差(记作 $IRMSE(t)_k$). 同样地,本文将网络流量样本 Y_t 其输入到 $NAES$ 中的每个局部自编码器中. 对于每个局部自编码器 NAE_k,它将输出一个重构样本 $\tilde{Y}_t$,并且本文计算 Y_t 与重构样本 $\tilde{Y}_t$ 之间的均方根误差(记作 $NRMSE(t)_k$). 随后,本文可以得到 IIoT 数据样本 S_t 的一个 RMSE 向量(记作 $ES(t)=<IRMSE(t)_1,IRMSE(t)_2,\cdots,IRMSE(t)_K,NRMSE(t)_1,NRMSE(t)_2,\cdots,NRMSE(t)_K>$),它表示了在不同语义空间中异常程度的分布情况. 最后,本文可以得到整个训练数据集 D 的 RMSE 向量集合(记作 $ES=\{ES(1),ES(2),\cdots,ES(|D|)\}$).

全局自编码器训练:全局自编码器(记作 GAE)被训练用于捕捉所有局部自编码器输出的正常模式. 具体而言,GAE 以 RMSE 向量 $ES(t)$ 作为输入,并通过编码器和解码器输出一个重构的 RMSE 向量 $E\tilde{S}(t)$. 本文在全局自编码器中使用多层感知机(MLP)作为全局编码器和解码器. 具体而言,本文采用两层的 MLP 结构,如式(9)所示,其中 $W^{(i)}$ 和 $b^{(i)}$ 分别是第 i 层隐藏层的可学习权重矩阵和偏置向量. 通过最小化式(10)中的损失函数,本文在 RMSE 向量数据集 ES 上训练 GAE.

$$\begin{aligned}h^{ES(1)}&=\mathrm{ReLU}(W^{(1)}\cdot ES(t)+b^{(1)})\\h^{ES(2)}&=\mathrm{ReLU}(W^{(2)}\cdot h^{ES(1)}+b^{(2)})\end{aligned}\tag{9}$$

$$Loss_G=\frac{1}{|D|}\sum_{t=1}^{|D|}\sqrt{\frac{\sum_{i=0}^{2K}(ES(t)[i]-E\tilde{S}(t)[i])^2}{2K}}\tag{10}$$

由于 GAE 是在所有数据模态和所有语义空间的输出上进行训练的,因此它有望能够发现那些基于单一自编码器或单一数据模态难以检测到的潜在异常. 例如,某些隐蔽的异常可能并不会导致所有局部自编码器都产生较高的重构 RMSE 值.

2.3.2　异常检测

对于一个 IIoT 数据样本 $S_t=(X_t,Y_t)$,本文首先将其输入到局部自编码器集合 $AES=\{IAE_1,IAE_2,\cdots,IAE_K,NAE_1,NAE_2,\cdots,NAE_K\}$ 中,这些局部自编码器将输出一组均方根误差 RMSEs,记为 $ES(t)=\{IRMSE(t)_1,IRMSE(t)_2,\cdots,IRMSE(t)_K,NRMSE(t)_1,NRMSE(t)_2,\cdots,NRMSE(t)_K\}$. 接着,本文将 $ES(t)$ 输入到全局自编码器 GAE 中,GAE 将输出一个重构后的 RMSE 集合 $E\tilde{S}(t)$. 本文计算 $ES(t)$ 与重构后的 RMSE 集合 $E\tilde{S}(t)$ 之间的全局均方根误差(记作 $GRMSE(t)$). 最后,将 $GRMSE(t)$ 与预定义的阈值 δ 进行比较. 如果 $GRMSE(t)>\delta$,则将 S_t 判定为异常,表明其可能存在与预期模式的偏离;否则,将 S_t 判定为正常.

3　实　验

3.1　实验准备

3.1.1　数据集

本文基于以下两个数据集对本文方法进行了评估,其中 HAI 数据集用于传感模型的消融实验.

ToN_IoT 数据集:该数据集[34]包含了来自传感器和网络流量的异构数据源. 这些数据是从由 UNSW Canberra Cyber 的物联网实验室设计的一个现实且大规模的测试平台网络中收集的. 具体来说,ToN_IoT 中的传感器子集包括从 7 个传感器中采样的 21 个数据维度,而网络流量子集则包含 46 个特征. 传感数据的采样频率为 1Hz. 正常事件与异常事件的比例为 24:1.

HAI 数据集:该数据集来自一个现实的 ICS 测试平台,配备了一个 HIL(硬件在环)模拟器,该模拟器模拟了蒸汽涡轮发电和抽水蓄能水电发电. 数据集的收集跨越了 11 天. 它包含了从 84 个传感器和执行器收集的每秒数据. 异常是由 50 次网络攻击生成的. 在 HAI 中,训练集和测试集明确分开,训练集在没有异常的情况下收集,测试集包含 1/40 的异常样本.

3.1.2　评价指标

由于异常事件的比例较低,数据集存在不平衡问题. 仅考虑准确率(Accuracy)是没有意义的,因为即使模型无法检测到异常事件,也可能因为大量正常样本的正确分类而获得较高的准确率. 因此,除了准确率之外,本文还使用了另外 3 个评估指标,即精确率(Precision)、召回率(Recall)和 F1 分数(F1-Score),如公式(11)、公式(12)和公式(13)所示. 其中,TP 表示正确检测到的异常样本数量,FP 表示被错误识别为异常的正常样本数量,FN 表示被错误识别为正常的异常样本数量.

$$Precision=\frac{TP}{TP+FP}\tag{11}$$

$$Recall=\frac{TP}{TP+FN}\tag{12}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

3.2 实验 1. 对比实验

为了评估本文方法的性能竞争力,本文将它与以下 7 种基线方法进行了比较. 所有这些基线方法都满足以下条件:a)采用无监督学习方式,不需要异常样本;b)同时考虑两种数据模态(即传感数据和网络流量数据);c)已经经过优化,以输出最佳性能.

阈值法(Threshold):该方法首先计算正常样本中每个属性的取值范围. 然后,对于一个 IIoT 数据样本,如果其任意属性值超出了对应的正常范围,则该样本将被识别为异常. 需要注意的是,传感样本的属性是不同传感器的维度,而网络流量样本的属性是从数据中提取的统计特征.

单类支持向量机(OC-SVM):一种基于单类支持向量机的异常检测模型[6]. 具体而言,它首先在高维特征空间中学习一个能够包含正常样本的边界,然后检测那些落在该边界之外的异常样本.

孤立森林(iForest):一种利用多棵决策树集成的异常检测模型[7]. 具体来说,它首先根据特征使用多棵决策树对样本进行分割,然后将平均路径长度较短的样本视为异常.

多层感知机自编码器(MLP-AE):一种基于自编码器的异常检测模型,它使用两层的多层感知机(MLP)作为编码器和解码器,并根据重构误差来识别异常[35].

长短期记忆自编码器(LSTM-AE):是一种基于自编码器的异常检测模型,它使用堆叠的两层长短期记忆网络(LSTM)作为编码器和解码器[14].

硬投票(Hard Voting):该方法首先为两种数据模态分别训练 2K 个局部自编码器. 然后,这些局部自编码器独立地进行异常检测决策. 最后,对于一个 IIoT 数据样本,如果超过 K 个局部自编码器判定其为异常,则该样本将被识别为异常.

软投票(Soft Voting):该方法同样首先为两种数据模态分别训练 2K 个局部自编码器. 然后,通过考虑局部自编码器的重构 RMSE 来做出最终的异常检测决策. 具体而言,对于一个 IIoT 数据样本,本文计算所有局部自编码器生成的重构 RMSE 的平均值,并将该平均重构 RMSE 与预定义的阈值 δ 进行比较.

在这些基线方法中,阈值法(Threshold)是一种基于规则的方法. 单类支持向量机(OC-SVM)、孤立森林(iForest)、多层感知机自编码器(MLP-AE)和长短期记忆自编码器(LSTM-AE)属于特征级融合方法. 具体来说,对于一个 IIoT 数据样本 $S_t = (X_t, Y_t)$,这些方法首先将多变量时间序列的传感样本 X_t 和网络流量样本 Y_t 的拼接成一个统一的向量 Z_t,并基于这些统一向量训练异常检测模型.

硬投票、软投票和本文方法属于模型级融合方法. 这些方法允许子模型独立生成输出,并基于全局策略对这些输出进行融合. 表 1 展示了比较结果,从结果中可以观察到以下趋势.

1)尽管阈值法(Threshold)是在工程实践中应用最为广泛的一种策略,但在本实验的数据集上,其对异常的检测性能却很低,尤其是对于那些传感器信号中没有突然变化的模式偏差异常. 通过对实验结果的分析,本文发现阈值法只能检测到物联网设备中的一些简单异常,例如温度的显著波动. 然而,数据集中的大多数异常是由复杂事件引起的,例如网络入侵. 例如,攻击者可能通过入侵破坏远程传感设备的运行,导致检测到的数据出现与正常模式的轻微偏差.

表 1 本文方法对比实验结果表

Table 1 Comparison experiment results of the method in this study

	准确率	精确率	召回率	F1
Threshold	0.726	0.996	0.329	0.495
OC-SVM	0.669	0.557	0.911	0.691
iForest	0.774	0.822	0.568	0.672
LSTM-AE	0.758	0.641	0.923	0.756
MLP-AE	0.678	0.558	0.998	0.716
Hard Voting	0.637	0.664	0.517	0.581
Soft Voting	0.815	0.801	0.821	0.810
本文方法	0.903	0.879	0.883	0.881

2)基于深度学习的方法(即 MLP-AE 和 LSTM-AE)比基于传统机器学习的方法(即 OC-SVM 和 iForest)表现更好. 这是因为深度学习方法具有更强大的学习能力,能够捕捉异构数据的潜在和语义特征,而传统机器学习方法仅提取非常表面的特征.

3)LSTM-AE 的性能优于 MLP-AE. 与 MLP 相比,LSTM 在对时间序列数据进行建模方面更为有效. 这表明 IIoT 数据样本中存在强烈的时间模式,尤其是在时间序列传感样本中.

4)硬投票的次优性能归因于其在检测复杂异常时的困难. 例如,由网络入侵引起的异常可能涉及与正常行为的细微偏差. 此外,由于每个自编码器的决策都被视为同等重要,如果异常只能被一小部分局部自编码器检测到,那么结果的聚合将是无效的.

5)在模型级融合方法中,本文方法优于硬投票和软投票. 这表明在子模型的输出上学习一个全局模型是一种比简单投票机制更好的模型融合策略. 尽管在无监督学习条件下,没有丰富的监督信号(例如分类概率分布、分类残差等)可供数据融合使用,但由多样化局部自编码器生成的重构误差仍可为模型级融合策略提供线索,以做出联合决策. 此外,本文方法在异常检测的整体性能上也表现最佳.

3.3 实验 2. 调参实验

本文方法中最重要的参数是 δ,即异常分数的阈值. 在这里,本文将均方根误差(RMSE)值归一化到[0,1]范围内,实验结果如图 5 所示. 首先,精确率和召回率呈现出相反的变化

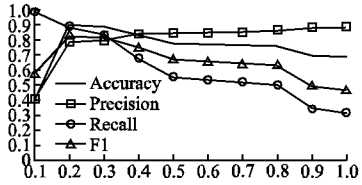

图 5 δ 的调参结果

Fig. 5 Effect of parameter δ

趋势. 随着 δ 的增加,模型对异常检测变得更加严格,因此检测到的异常数量会减少. 结果是,精确率呈现出稳定的上升阶段,而召回率呈现出稳定的下降阶段. 其次,准确率和 F1 分数

在将 δ 从 0.1 增加到 0.2 时表现出显著的上升趋势,而当进一步增加 δ 时,则呈现出稳定的下降趋势. 通过将 δ 设置为 0.2,可以实现最佳的整体性能.

3.4　实验 3. 消融实验

为了研究两种数据模态(即传感数据和网络流量数据)对异常检测性能的影响,本文分别对基于传感数据的自编码器(称为 IoT-AE)和基于网络流量数据的自编码器(称为 Network-AE)进行了单独测试. 在这里,IoT-AE 是根据第 2.3.1 节的描述作为全局自编码器进行训练的,训练时不包括网络流量数据的局部自编码器. 同样地,Network-AE 是作为全局自编码器进行训练的,训练时不包括传感数据的局部自编码器.

在第 1 个实验中,本文将 IoT-AE 的性能与本文方法以及一些其他基线方法进行了比较,这些基线方法如下. 所有这些基线方法仅考虑传感数据.

MLP-IAE:这是一种基于传感数据的自编码器异常检测模型. 它使用两层的多层感知机(MLP)作为编码器和解码器.

LSTM-IAE:这也是一种基于传感数据的自编码器异常检测模型. 它使用两层堆叠的长短期记忆网络(LSTM)作为编码器和解码器.

实验结果如表 2 所示. 首先,LSTM-IAE、MTGNN 和 IoT-AE 的性能显著优于 MLP-IAE. 这表明时间模式对于基于传感数据的异常检测至关重要,不应被忽视. 其次,IoT-AE 相较于 LSTM-IAE 有轻微的优势. 这说明利用图来捕捉多变量时间序列数据不同维度之间的空间相关性,对于异常检测是有帮助的.

表 2　基于传感数据的自编码器的效果评估

Table 2　Evaluation of the sensing data based autoencoder

	准确率	精确率	召回率	F1
MLP-IAE	0.721	0.593	0.833	0.692
LSTM-IAE	0.852	0.772	0.903	0.832
IoT-AE	0.844	0.823	0.856	0.840
本文方法	0.904	0.881	0.882	0.880

在第 2 个实验中,本文将 Network-AE 的性能与本文方法以及一些其他基线方法进行了比较,这些基线方法如下. 所有这些基线方法仅考虑网络流量数据,并且本文根据第 2.2.2 节的描述提取统计特征作为这些基线方法的输入.

iForest:这是仅基于网络流量数据训练的 iForest 模型.

MLP-NAE:这是一种仅基于网络流量数据的自编码器异常检测模型. 它使用两层的多层感知机(MLP)作为编码器和解码器.

表 3　基于网络流量数据的自编码器的效果评估

Table 3　Evaluation of the network traffic data based autoencoder

	准确率	精确率	召回率	F1
iForest	0.751	0.799	0.517	0.628
MLP-NAE	0.762	0.635	0.974	0.769
Network-AE	0.901	0.967	0.784	0.865
本文方法	0.903	0.879	0.883	0.881

实验结果如表 3 所示. 首先,基于深度学习的模型(即 MLP-NAE 和 Network-AE)优于基于传统机器学习的模型(即 iForest),这也在第 3.2 节的先前实验中得到了验证. 其次,Network-AE 的性能优于 MLP-NAE,这表明模型集成的优势,它可以降低过拟合的风险. 最后,本文方法表现最佳,这体现了多视图数据融合的优势.

4　结　论

在本文中,本文研究了工业物联网(IIoT)系统中的异常检测问题. 本文提出了一个新颖的无监督深度学习框架,通过融合多个自编码器来识别异常,同时综合考虑了 IIoT 系统中的传感数据和网络流量数据. 通过挖掘和学习多种数据模态之间的相关性模式,本文方法在 ToN_IoT 数据集上的表现优于其他最先进的模型.

未来的研究可以从两个方面展开. 首先,本文方法目前只能检测异常事件,而无法识别这些异常事件的具体类型. 因此,增强本文方法的异常分类能力将有助于更好地应对这些异常事件. 其次,如何对检测到的异常事件进行诊断以及追溯其根源,也是值得进一步研究的方向.

References:

[1] DING X O, YU S J, WANG M X, et al. Anomaly detection on industrial time series based on correlation analysis[J]. Journal of Software, 2020, 31(3): 726-747.

[2] ZHENG Y J, HE Q, ZHANG C L, et al. GRU-attention based unsupervised multivariate time series anomaly detection[J]. Journal of Shanxi University, 2020, 43(4): 756-764.

[3] LI Y F, LI M Y, CHANG X, et al. Anomaly detection of IoT water quality monitoring data based on explainable deep learning[J]. Computer Science, 2024, 50(6): 179-187.

[4] Singhal A, Seborg D E. Clustering multivariate time-series data[J]. Journal of Chemometrics: a Journal of the Chemometrics Society, 2005, 19(8): 427-438.

[5] SHEN Y T, TAN Y H, XIA W C. Research on anomaly detection algorithm of internet of things equipment[J]. Intelligent Computer and Applications, 2024, 14(8): 225-233.

[6] Erfani S M, Rajasegarar S, Karunasekera S, et al. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning[J]. Pattern Recognition, 2016, 58(1): 121-134.

[7] Thornton G, Zadeh P B. An investigation into Unmanned Aerial System(UAS) forensics: data extraction & analysis[J]. Forensic Science International: Digital Investigation, 2022, 41(1): 1-22.

[8] Malhotra P, Ramakrishnan A, Anand G, et al. LSTM-based encoder-decoder for multi-sensor anomaly detection[J]. arXiv preprint arXiv: 1607.00148, 2016.

[9] Akçay S, Atapour-Abarghouei A, Breckon T P. Skip-ganomaly: skip connected and adversarially trained encoder-decoder anomaly detection[C]//International Joint Conference on Neural Networks(IJCNN), 2019: 1-8.

[10] Nanduri A, Sherry L. Anomaly detection in aircraft data using Recurrent Neural Networks(RNN)[C]//Integrated Communications Navigation and Surveillance(ICNS), 2016: 5C2-1-5C2-8.

[11] Ambusaidi M, He X, Nanda P, et al. Building and intrusion detection system using a filter-based feature selection algorithm[J]. IEEE Transactions on Computer, 2016, 65(10): 2986-2998.

[12] LENG Q J. Research on anomaly detection algorithm of network traffic based on deep learning[J]. Heilongjiang Science,2025,16(2):127-129.
[13] Xu W,Jang Jaccard J,Singh A,et al. Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset[J]. IEEE Access,2021,9(1):140136-140146.
[14] Said Elsayed M,Le Khac N A,Dev S,et al. Network anomaly detection using LSTM based autoencoder[C]//Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks,2020:37-45.
[15] Griffin J M,Doberti A J,Hernández V,et al. Multiple classification of the force and acceleration signals extracted during multiple machine processes: part 1 intelligent classification from an anomaly perspective[J]. The International Journal of Advanced Manufacturing Technology,2017,93(1):811-823.
[16] Janssens O,Slavkovikj V,Vervisch B,et al. Convolutional neural network based fault detection for rotating machinery[J]. Journal of Sound and Vibration,2016,377(1):331-345.
[17] Amruthnath N,Gupta T. A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance[C]//5th International Conference on Industrial Engineering and Applications(ICIEA),2018:355-361.
[18] Diez Olivan A,Pagan J A,Khoa N L D,et al. Kernel-based support vector machines for automated health status assessment in monitoring sensor data[J]. International Journal of Advanced Manufacturing Technology,2018,95(1):327-340.
[19] Joshi S S,Phoha V V. Investigating hidden Markov models capabilities in anomaly detection[C]//Proceedings of the 43rd Annual ACM Southeast Conference,2005:98-103.
[20] Li C L,Sohn K,Yoon J,et al. Cutpaste: self-supervised learning for anomaly detection and localization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,2021:9664-9674.
[21] Kingma D P,Welling M. Auto-encoding variational bayes[EB/OL]. http://web2. cs. columbia. edu/~blei/fogm/2018F/materials/kingmaWelling 2013. pdf,2013-12-20.
[22] Lu C,Wang Z Y,Qin W L,et al. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification[J]. Signal Processing,2017,130(1):377-388.
[23] Zhang C,Song D,Chen Y,et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data[C]//Proceedings of the AAAI Conference on Artificial Intelligence,2019:1409-1416.
[24] Yin C,Zhang S,Wang J,et al. Anomaly detection based on convolutional recurrent autoencoder for IoT time series[J]. IEEE Transactions on Systems,Man,and Cybernetics: Systems,2020,52(1):112-122.
[25] Muneer A,Taib S M,Fati S M,et al. A hybrid deep learning-based unsupervised anomaly detection in high dimensional data[J]. Computers,Materials & Continua,2022,70(3):5363-5381.
[26] Zheng Y. Methodologies for cross-domain data fusion: an overview[J]. IEEE Transactions on Big Data,2015,1(1):16-34.
[27] Xiao X,Zheng Y,Luo Q,et al. Inferring social ties between users with human location history[J]. Journal of Ambient Intelligence and Humanized Computing,2014,5(1):3-19.
[28] Ngiam J,Khosla A,Kim M,et al. Multimodal deep learning[C]//International Conference on Machine Learning,2011:689-696.
[29] Nagrani A,Yang S,Arnab A,et al. Attention bottlenecks for multimodal fusion[J]. Advances in Neural Information Processing Systems,2021,34(1):14200-14213.
[30] Sun Y,Qi J,Zheng Y,et al. K-nearest neighbor temporal aggregate queries[C]//Proceedings of the 18th International Conference on Extending Database Technology,2015:493-504.
[31] Ngo G,Beard R,Chandra R. Evolutionary bagging for ensemble learning[J]. Neurocomputing,2022,510(21):1-14.
[32] Kadkhodaei H R,Moghadam A M E,Dehghan M. HBoost: a heterogeneous ensemble classifier based on the Boosting method and entropy measurement[J]. Expert Systems with Applications,2020,157(1):1-14.
[33] Zhang H,Li J L,Liu X M,et al. Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection[J]. Future Generation Computer Systems,2021,122(1):130-143.
[34] Booij T M,Chiscop I,Meeuwissen E,et al. ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets[J]. IEEE Internet of Things Journal,2021,9(1):485-496.
[35] Al Safaar D,Al Yaseen W L. Hybrid AE-MLP: hybrid deep learning model based on autoencoder and multilayer perceptron model for intrusion detection system[J]. International Journal of Intelligent Engineering & Systems,2023,16(2):35-49.

附中文参考文献：

[1] 丁小欧,于晟健,王沐贤,等. 基于相关性分析的工业时序数据异常检测[J]. 软件学报,2020,31(3):726-747.
[2] 郑育靖,何 强,张长伦,等. 基于 GRU-Attention 的无监督多变量时间序列异常检测[J]. 山西大学学报(自然科版),2020,43(4):756-764.
[3] 李永飞,李铭洋,常 鑫,等. 基于可解释性深度学习的物联网水质监测数据异常检测[J]. 计算机科学,2024,50(6):179-187.
[5] 申煜铜,谈宇浩,夏文超. 基于联邦学习的物联网设备异常检测算法研究[J]. 智能计算机与应用,2024,14(8):225-233.
[12] 冷秋君. 基于深度学习的网络流量异常检测算法研究[J]. 黑龙江科学,2025,16(2):127-129.