# 南 开 大 学

## 本科生毕业论文（设计）

中文题目：面向多元时间序列预测的后门攻击

外文题目：Backdoor Attacks to Multivariate Time Series Forecasting


学　　号：2110813

姓　　名：陈希

年　　级：2021 级

专　　业：计算机科学与技术

系　　别：计算机科学与技术

学　　院：计算机学院

指导教师：蔡祥睿　副教授


完成日期：2025 年 10 月

# 关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年　　月　　日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年　　月　　日

# 摘　要

随着深度学习模型的发展，时间序列预测在众多领域都发挥出了关键作用，广泛地应用于经济，气象，医学，工业等场景。然而时间序列预测模型也面临着后门攻击的威胁。通过在模型训练时注入后门，让模型的输出预测能够达到攻击者的意图。然而，真实世界的时间序列普遍存在缺失值，这为攻击与防御机制的设计提出了新的挑战。现有后门攻击方法大多假设数据完整，忽略了缺失值信息及其对攻击策略设计的影响。基于上述内容，本文的核心研究内容如下：

针对于目前后门攻击基本基于完整时间序列的问题，本文提出了一种新颖的后门攻击框架，更加关注缺失值的信息，首先设计了一种结合重要时间步分析的插补策略，通过平衡时间一致性与插补平滑性，恢复数据完整性，并且通过实验对比窗口均值插补方案验证了插补策略的有效性和必要性。同时将触发器的注入位置与时间序列的内在特性紧密耦合，优先选择靠近缺失数据段和模型预测关键时间步的位置进行注入，结合模型预测的关键时间步重要性以及局部数据变化率生成综合得分，从而能够让触发器设置在模型可能更不稳定、更容易受扰动影响的地方以及注入行为本身可能因插补数据的存在而更不易被察觉的位置，提高了后门攻击的隐蔽性。为了保证触发器能够更加有效地实现攻击效果，本文设计了一种强关联学习机制，强制模型学习从注入触发器的输入到目标模式的映射关系，鼓励模型形成识别触发器-目标模式的特征。

实验结果表明，在 PeMS 交通数据集上，本文方法的攻击性能相比于 Clean 的干净预测基准方法，攻击平均绝对误差 $MAE_a$ 降低了接近 45%，并且对比 Clean，Random 和 Manhattan 三种基线方法的攻击效果，$MAE_a$ 取得了从 17% 到 43% 的不同程度的相对优化效果，同时在隐蔽性方面基本接近随即猜测结果，验证了本文后门攻击方法的有效性和隐蔽性。

**关键词**：时间序列；缺失值插补；后门攻击；触发器

# Abstract

With the advancement of deep learning models, time series forecasting has become pivotal across numerous domains, including economics, meteorology, medicine, and industry. However, these forecasting models also face the threat of insidious backdoor attacks, where malicious actors inject backdoors during training to manipulate model predictions towards attacker-intended outcomes. A significant real-world challenge is the prevalence of missing values in time series data, which poses new complexities for both attack and defense mechanism design. Existing backdoor attack methods predominantly assume data completeness, neglecting missing value information and its profound implications for attack strategy formulation. Addressing this gap, this paper proposes the following core research contributions:

To tackle the prevalent issue of backdoor attacks operating primarily on complete time series, we introduce a novel backdoor attack framework that more acutely considers information pertaining to missing values. Our approach first incorporates an imputation strategy, designed by integrating time step importance analysis. This strategy meticulously balances temporal consistency with imputation smoothness to restore data integrity, validated through comparative experiments against a window-mean imputation scheme, underscoring its effectiveness and necessity.Crucially, our attack strategy tightly couples trigger injection locations with the intrinsic characteristics of the time series. We prioritize injecting triggers in proximity to missing data segments and time steps identified as critical for model prediction. A composite score, generated by combining the density of missing values within the local injection window, the importance of critical predictive time steps, and local data volatility, guides trigger placement. This strategic positioning aims to site triggers where the model may exhibit greater instability and susceptibility to perturbations, or where the injection itself might be better concealed by the presence of imputed data, thereby enhancing the backdoor attack's stealthiness. Furthermore, to ensure the trigger robustly achieves the desired attack effect, we design a strong association learning mechanism. This mechanism explicitly compels the model to learn the mapping from trigger-injected inputs to the target pat-

tern, thereby encouraging the formation of discriminative features for the trigger-pattern pair.

Experimental results on PeMS traffic datasets demonstrate that, compared to the clean prediction baseline (Clean), our proposed method reduces $MAE_a$ by approximately 45%. Moreover, when benchmarked against three baseline attack methods (Clean, Random, and Manhattan), our approach achieves relative $MAE_a$ improvements ranging from 17% to 43%. Concurrently, the stealthiness metrics of our method are comparable to those of random guessing, validating the effectiveness and covertness of the proposed backdoor attack methodology.

**Key Words:** Time series;Missing value imputation;Backdoor attack;Trigger

# 目 录

# 第一章　绪论

## 第一节　研究背景和意义

随着物联网技术的快速发展和传感器网络的普及，时间序列数据已成为数字化社会中最重要的数据形态之一。从金融市场[1]的高频交易数据到智能电网的实时能耗监测，从可穿戴设备的生物信号采集到城市交通流量[2-3]的动态感知，时间序列数据以其特有的时序依赖性和动态演化特征，为各领域的智能决策系统提供了关键支撑。随着物联网设备的激增和数字化转型的加速，全球产生的时间序列数据量正以前所未有的速度增长[4-5]，显著高于其他数据类型，这凸显了时间序列分析技术在当代智能系统中的核心地位。

在时间序列预测技术演进过程中，深度学习模型展现出了革命性的突破。前几年有以长短期记忆网络（Long Short-Term Memory，LSTM）[6]为代表的循环神经网络通过门控机制有效捕捉长期依赖关系；Transformer[7]架构凭借自注意力机制在多元时间序列预测中取得突破性进展；而图神经网络（Graph Neural Networks，GNN）[8]则通过建模时空相关性在交通流量预测等领域表现卓越。工业界的应用数据表明，在电力负荷预测场景中，深度学习方法较传统 ARIMA 模型[9]将预测精度提升了 40% 以上，充分验证了其技术优势。

然而，深度学习模型的广泛应用也伴随着新的安全风险，后门攻击就是其中一种极具威胁的攻击形式。后门攻击作为一种隐蔽的训练时攻击，早已在图像识别[10] 和自然语言处理[11] 等领域被广泛研究。攻击者通过在训练数据中植入精心设计的“触发器”，使得被感染的模型在部署后，遇到包含触发器的输入时会产生攻击者预设的恶意输出，而在处理正常输入时则表现得与良性模型无异。这种攻击的隐蔽性使得传统基于测试准确率的模型验证方法难以发现其存在。

近年来，后门攻击的威胁开始向时间序列领域蔓延，但相关研究仍相对匮乏。已有的少数针对时间序列的后门攻击研究，如 BadNet-TS[10]，主要关注在数据完整的理想场景下，通过在固定位置叠加静态触发模式来实现攻击。然而，现有研究普遍忽视了一个关键且广泛存在的问题——时间序列数据的缺失性。在现实世界中，由于传感器故障、网络传输丢包、存储异常等原因，超过 60% 的工业时间序列数据[12]存在不同程度的缺失。这种数据的普遍不完整性，使得那

些基于完整数据假设的后门攻击方法在实际应用中并不能实现很好的攻击效果，甚至无法成功地应用于这些场景中。

从目前学术界的研究方向来看，缺失值处理和对模型安全的研究往往是割裂的。一方面，数据挖掘和机器学习领域致力于发展更先进的时间序列插补技术，例如基于生成对抗网络（GAN）的 C-RNN-GAN[13]，通过对抗训练来对序列的整体联合概率建模并生成高质量的数据序列，鼓励模型学习更多变的模式并且生成更加接近真实场景的数据序列；基于 Transformer 的 SAITS[14]，从两个对角屏蔽自注意力块的加权组合中学习缺失值，显式捕获时间步之间的时间依赖性和特征相关性，在现实世界中不完整时间序列数据上表现出了优于最先进插补方法的性能。另一方面，安全领域的研究主要集中在完整数据条件下的后门攻击检测与防御机制上。基于模型内部检查的检测方法: 这类方法试图通过分析模型自身的参数或中间层输出来识别异常。例如，一些研究[15-16] 尝试逆向工程触发器模式，通过优化输入来寻找能最大化特定类别激活或导致模型行为剧变的最小输入扰动，从而识别可疑的后门触发器。另一些工作通过关注神经元激活模式[17]，假设被感染的模型在处理包含触发器的输入时会表现出与处理干净输入时不同的内部激活特征来区分良性模型和后门模型。

这些研究上的割裂忽视了一个潜在的问题：这种割裂性研究范式忽略了一个潜在风险：缺失值处理流程可能被攻击者恶意利用。攻击者能否利用对缺失模式的了解，甚至潜在地影响插补过程，来设计与数据缺失状态相协同的动态触发器？能否在插补后的数据中植入与原始数据模式更契合、更难被发现的后门？这些在存在缺失值场景下的时间序列后门攻击问题，是当前研究的一个显著空白。

因此，本文提出一个将时间序列插补与后门攻击相结合的框架，首先通过基于重要时间步分析的插补器处理缺失值，然后在插补后的数据上进行攻击注入和优化。同时设计一种基于时空关联性和与目标模式强关联性的动态触发器生成机制，鼓励模型利用局部数据特性和时间步重要性，生成更加隐蔽并且以目标模式为导向的触发器，优先关注在缺失值附近及重要时间步周围区域注入。最后，结合多损失函数的优化，集合各个任务全面地进行优化，从而在各个任务上能够取得优化效果。

## 第二节　国内外研究现状

### 1.2.1　时间序列预测

时间序列预测作为数据分析的核心任务之一，在众多领域扮演着至关重要的角色。其核心挑战在于有效捕捉复杂的时间依赖关系，并准确预测未来的序列动态。近年来，深度学习模型极大地推动了时间序列预测模型的发展，涌现出多种具有代表性的架构。

作为处理序列数据的经典方法，循环神经网络以及其变种，在早期时间序列预测任务[18]上取得了一定的成功。例如长短期记忆网络[6]和门控循环单元（Gate Recurrent Unit，GRU）[19]通过引入门控机制来控制信息的流动和记忆的更新，有效缓解了传统循环神经网络（Recurrent Neural Network，RNN）在处理长序列时遇到的梯度消失和梯度爆炸问题，使得模型能够学习到更长期的依赖关系，从而在长时间序列预测任务上展现出了优秀的性能。

然而，因为对于多元时间序列，简单 RNN 难以显式捕捉变量间的复杂交互关系。而又因为借鉴到了 Transformer 在自然语言处理领域的巨大成功，于是 Transformer 架构被引入时间序列预测[7]，通过自注意力机制得模型能够直接捕捉任意两个时间步之间的依赖关系而不受距离限制。同时，自注意力的计算可以高度并行化，大大提高了训练效率。而后针对 Transformer 应用于长时间序列的挑战，例如自注意力机制带来的计算和内存复杂度，Li 等人[20]通过引入卷积自注意力层和局部注意力机制来降低复杂度；Zhou 等人[21]提出了 ProbSparse 自注意力机制，通过计算最重要的少数几个 Query-Key 对来近似全局注意力，并且设计了自注意力蒸馏操作，在编码器层级逐步缩短序列长度，减少计算量和内存占用，从而让 Transformer 在超长时间序列预测任务上也能够发挥出很好的优势。

在 Transformer 奠定的基础上，为了提升模型处理时间序列复杂特性的效率和精度，研究者们从不同角度对架构进行了更深入的创新。Autoformer[22]创新了传统的分解序列作为预处理的方式，对时间序列进行深度分解，通过自相关机制代替自注意力机制，使用快速傅里叶变换计算时间滞后序列间的相似性来发现周期性依赖，同时使用累计分解，使得 Autoformer 能更清晰地建模时间序列的不同组成部分，并在效率和精度上取得了显著提升。继承了 Autoformer 的分

解思想，阿里达摩院设计了 FEDformer[23] 追求时间序列在频域中更稀疏的表示，以实现更高的计算效率和对噪声分布变化的鲁棒性。它同样在模型内部进行序列分解，但在处理分解后的项时，引入了傅里叶增强变换和小波增强变换，从而更好地捕捉主要变化模式并抵抗噪声。而后，清华大学提出了 TimesNet 模型[24]，从时间序列固有的多周期性出发，例如日、周、季节等多种相互交织的周期模式，通过使用快速傅里叶变换（Fast Fourier Transform，FFT）分析输入的一维时间序列，将原始的一维序列重塑成多个二维张量，并且使用二维卷积特征提取特征，捕捉周期内部的变化模式和不同周期之间的变化趋势。实验证明，TimesNet 不仅在长期和短期预测任务上表现出色，在时间序列插补、分类和异常检测等多个相关任务上也达到了顶尖水平，展现了其作为时间序列分析基础模型的潜力。

时间序列预测的研究并未局限于 Transformer 的演化，而是在越来越多的方向进行探索。S4 模型[25]通过重新参数化 SSM（State Space Model，SSM）状态矩阵解决了长时间序列计算带来的内存和时间复杂度问题，实现了一个能够高效处理长距离依赖的通用序列模型，并且后续再次创新，将动态选择机制与状态空间模型相结合，提出了选择性 SSM 的概念[26]，将状态更新过程块状化并行处理。此外，对于具有明确空间结构的数据，例如交通网络数据，Yu 等人[27]提出了时空图神经网络，结合了 TemporalConv 和 SpatialConv 进行时间和空间特征提取，克服了 RNN 在序列学习中积累误差的挑战，并且建模空间依赖性。

### 1.2.2 时间序列缺失值插补

完整的时间序列一直是时间序列预测任务的重要条件，因此，目前的大部分研究都是基于完整的时间序列展开的，但是在实际的应用场景中会遇到许多存在缺失值的时间序列，例如在医学领域，数据收集可能会遭遇用户隐私问题，有些用户不愿意公开自己的病例数据，导致数据缺失。因此，得到一个更加接近真实数据的插补值，对于存在缺失值场景的时间序列预测任务十分重要。

现有研究表明，大致可以分为基于统计学的数据插补和基于机器学习，深度学习的数据插补方法。从基于统计学的缺失值插补方法来看，通过前推，后推，中值和均值插补，能够满足对于缺失值精度要求不高的场景，例如工业监测中，往往数据都趋于平缓，缺失率较低，因此通过均值插补基本就可以满足需求。此

外，热卡填充[28]也提供了一种插补的新思路，对于一个包含空值的序列，热卡填充法在完整数据中找到一个与它最相似的序列，然后用这个相似序列的值来进行填充，但是这种方式的相似的标准很难界定。于是有了从拟合方法的缺失值插补方法[29]，主要思路是对于带有缺失值的序列，将已知数据集带入回归方程来估计预测值，并以此预测值来进行填充，但是当序列值非线性时，预测效果可能会有一定程度的下降。多重插补方法一定程度上解决了单重插补的局限性，通过计算均值，中位数，众数等数据，为每一个缺失值都产生一套可能的插补值，然后对插补集合根据评分函数进行选择，产生最终的插补值，但是该方法对于存在大范围缺失值的样本插补效果不佳。

相较于传统的基于统计学的方法，基于机器学习和深度学习的方法能够更加准确地捕捉时序关系，改善预测效果。Che 等人[30]对标准的 GRU 进行了改进，更加着重地考虑了缺失模式，通过引入了时间衰减机制，关联输入变量和隐藏状态，削弱距离当前时间较远的观测点的影响权重，并且将缺失掩码作为网络输入的一部分，让模型能够学习到缺失状态下的信息，这也是早期专门为处理含缺失值的多元时间序列设计的经典 RNN 模型之一。Su 等人[31]提出了一种非线性补偿算法来处理缺失值。该方法通过针对于多维度的时间序列变量之间的相互依赖关系进行插补，从而能够捕捉到局部和全局的特征，为复杂的多维时间序列数据提供了一种新颖的解决方案。Wang 等人[32]提出了一种改进的生成对抗网络用于缺失数据的插补，名为 PC-GAIN 模型，C 代表生成器 G 不仅接收噪声，还接收观测到的数据部分作为条件，来生成缺失部分；P 代表利用自身对缺失值的高置信度预测来辅助训练，提升模型在只有部分数据标签或无真实缺失值标签情况下的学习效果和插补性能。此外，Tang 等人[33]提出了一种 Memory Module 的参数化的特征捕捉模块，通过捕捉全局时间序列的特征，随后将局部特征与 Memory Module 进行相似度计算，考虑局部特征和全局特征的相似性，结合全局和局部模式计算得出插补值，通过一个统一的框架，同时学习和融合这两种不同尺度的时间动态。

### 1.2.3 后门攻击

后门攻击是针对机器学习和深度学习模型[34]训练时进行攻击，即当模型进行训练过程时，在训练数据集中注入后门，这个后门能够让模型在干净的数据集上表现正常，难以察觉；而当遇到含有触发器的数据集时，模型能够产生攻击者预设的目标模式。

2017 年，Gu 等人[10]提出了 BadNets，这也是最早提出的标志性后门攻击之一。该方法通过向训练集中注入带有小块像素图案的触发器并标记为目标类别的样本来植入后门。类似于 BadNets，Liu 等人[35]也通过数据投毒，作者意图通过额外数据重训练模型，使其在正常情况下表现正常，而遇到触发器时，表现出预设的行为。而由于神经网络的不可解释性[36]，也为触发器的隐蔽性提供了保证。

在触发器的设计方面，Chen 等人[37]通过使用人眼难以察觉的微小扰动作为后门触发器，也开启了对于触发器的隐蔽性的探索，在频域方向，Li 等人[38]指出大多数后门攻击的触发器集合都是静态触发集，从而导致了模型的泛化能力较差，因此作者提出了在图像频域嵌入触发器，提高隐蔽性。Barni 等人[39]较早地探索了通过语义含义的触发器进行攻击，并且采用干净标签策略，有毒数据集上的的真实标签与目标模式设定的标签保持一致，从而降低了模型发现后门攻击的能力。Nguyen 等人[40]创新了触发器的生成方式，根据输入样本动态生成触发器，设计一个专门生成触发器的生成器。

在跨领域后门攻击方面，在自然语言处理领域，Qi 等人[11]提出了使用隐蔽性高的句法结构作为触发器来进行后门攻击，攻击者选择一些输入样本注入触发器，并且将这些中毒样本标签设置为目标标签，随后一起训练，并且考虑了在特定数据集上微调模型，应对特定目标任务，实现了在所有受害者模型上都达到很高的攻击成功率。在 GNN 领域，Xi 等人[41]研究了针对图神经网络的后门攻击，通过使用特定子图结构作为触发器，并且不同的图产生不同的触发器，而又由于 GNN 在归纳式和直推式设置下都很易受后门攻击，因此在针对归纳式和直推式任务中适用性很高。

在时间序列领域，后门攻击主要可以分为时间序列分类的后门攻击和时间序列预测的后门攻击。时间序列分类（TSC）任务目标是将整个时间序列片段赋

予一个预定义的类别标签。而针对时间序列分类的后门攻击是指在不影响模型对良性样本分类准确率的前提下，使得模型在遇到嵌入了特定触发器的时间序列时，将其错误地分类到攻击者指定的目标类别。Ding 等人[42]提出一种后门攻击框架 TimeTrojan，旨在通过约束多目标优化学习形成触发模式，使得带有触发器的样本始终在原样本流形空间上，并且不再将触发器局限于某个具体的位置，而是鼓励触发器注入在能够影响模型预测最大的位置，从而改变模型的预测结果。在时间序列预测的后门攻击领域，Lin 等人[43]提出了 BACKTIME 的时间序列后门攻击方案，通过双层循环最小化模型输出和目标模式之间的损失并且基于历史数据生成触发器，触发器与目标模式进行强关联，从而能够让模型能够捕捉到触发器和目标模式之间的联系，更好地学习触发器模式，在多种受害者模型上都取得了显著的效果。

现有研究表明，对于时间序列的预测，缺失值的插补，图像分类和图像分类的后门攻击的成果颇丰，但是针对缺失值场景下的时间序列后门攻击的工作相对匮乏，本节总结了关于这个方向需要解决的问题。

### 1.2.4 总结与分析

从目前的工作来看，基本上大部分工作都是着重于缺失值的插补和触发器的生成中的一个方向，而很少将两者结合起来考虑，而由于大部分的研究都是基于完整的时间序列这一前提条件，因此很难直接在缺失数据上进行处理，因此需要考虑如何在触发器的生成策略上能够结合缺失值的相关信息，例如缺失值的密度，缺失值的位置等，对于缺失值的插补，也需要考虑到后续的后门攻击任务时需要一个怎样精度的缺失值。同时，触发器和目标模式的生成模式和插入位置也十分值得思考，选取一个让模型能够更加容易学习到的位置进行触发器注入，能够提高后门攻击性能，更容易被学习到的目标模式也能让模型在输出时更加接近攻击者意图。

此外，触发器的有效性和隐蔽性之间的平衡一直是一个工作重点，因为两者很难兼顾，若触发器隐蔽性较高，那么模型较难学习到触发器的特征，则会导致触发器有效性较低，而降低了触发器的隐蔽性，则会导致触发器容易被异常检测或者人工审查发现，从而导致了后门攻击的失败，因此在这两者之间取得平衡对

于成功的后门攻击极为重要。

## 第三节　论文组织结构

本文总共分为五个章节，整体组织结构如下：

第一章：绪论。主要阐述了关于时间序列预测以及后门攻击的相关工作，研究现状，对于目前存在的问题以及大致的解决思路进行了一个总体介绍，引出本文的主要工作内容，并且对文章结构进行一个说明。

第二章：相关技术介绍。根据本文工作内容，对于时间序列缺失值插补和后门攻击的一些概念进行解释和阐述，并且对于后续工作中使用的一些神经网络，深度学习模型等进行介绍。

第三章：方法构建和实验设计。这一章会详细地介绍本文的工作，围绕重要时间步分析算法展开，通过结合重要时间步分析进行缺失值的插补。通过多层感知机进行触发器的生成，捕捉变量之间的关系，同时设计了两种目标模式，与触发器之间形成触发器-目标模式的强关联模式，最后，结合多目标的损失函数，考虑泛化能力和攻击效果，关注触发器有效性和隐蔽性，进行整体优化。

第四章：实验数据和分析总结。这一章会详细地分析实验设置和实验数据，将干净数据集上的预测能力以及后门攻击效果与其他几种触发器设置模式进行对比，验证了本文方法的有效性，在攻击性能上整体高于基线。并且展开了隐蔽性评估和消融实验，分析了空间和时间注入率的影响和原因。

第五章：实验总结和未来展望。结合所有得到的数据和分析，基于目前的实验结果，对未来可能的工作和优化内容进行展望。

# 第二章　相关研究基础

## 第一节　时间序列

时间序列（Time Series）是指按照时间顺序排列的观测数据集合，其数据点通常以固定时间间隔，如秒、日、月、年记录，反映了某一现象或指标随时间变化的动态特征。时间序列广泛存在于[44]经济、[45]气象、[46]工业、[47]医疗等领域，例如股票价格、气温变化等都属于典型的时间序列数据。时间序列的核心特征包括趋势性、季节性、周期性和随机性等组成部分。

时间序列通常也可以分为单变量时间序列（univariate time series）和多变量时间序列（multivariate time series）。**单变量时间序列**是由单个变量随时间推移产生的一系列观测值。它记录了一个特定指标在连续时间点上的变化，其数学表示是：

$$\mathbf{x} = (x_1, x_2, \ldots, x_T) \tag{2.1}$$

其中 $x_t$ 表示在离散时间点 $t$ 观测到的标量值，$T$ 表示该时间序列的总长度。

**多变量时间序列**也可以称作多维时间序列，包含多个相关变量的联合观测，各个变量之间存在一定的联系，例如在记录车流信息时，会记录车流量，车速等信息，其数学表示是：

$$\mathbf{x}_t = (x_t^1, x_t^2, \ldots, x_t^M)^\top \in \mathbb{R}^M, \quad t = 1, 2, \ldots, T \tag{2.2}$$

其中 $x_t^m$ 表示在时间点 $t$ 观测到的第 $m$ 个维度的标量值，$T$ 表示时间序列的总长度，$M$ 表示变量的数量，即维度或特征数。

时间序列预测（Time Series Forecasting）是基于历史观测数据，通过建立数学模型或算法，对未来某一时间点的数值或序列整体变化趋势进行估计的统计分析方法。其核心假设是数据的历史模式在未来仍会持续，并通过捕捉这些规律实现预测。时间序列预测的应用场景丰富，例如商品销量预测、金融风险分析以及工业设备故障预警等。时间序列预测也可以分为单变量时间序列预测和多变量时间序列预测两种。其预测方式都可以解释为利用历史 $p$ 步预测未来 $h$ 步。

对于单变量时间序列预测，其数学公式可以表示为：

$$\hat{x}_{T+h} = f\left(x_T, x_{T-1}, \ldots, x_{T-p+1}\right) \tag{2.3}$$

其中 $f(\cdot)$ 表示预测模型，$\hat{x}_{T+h}$ 表示未来第 $h$ 步的预测值。

对于多变量时间序列预测，其数学公式可以表示为：

$$\hat{\mathbf{x}}_{T+h} = f\left(\mathbf{x}_T, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_{T-p+1}\right) \tag{2.4}$$

若输出为单变量，则 $\hat{\mathbf{x}}_{T+h}$ 退化为标量 $\hat{x}_{T+h}$.

## 第二节　时间序列缺失值插补

时间序列的缺失是指一系列的时间序列中因各种原因导致的一个或者一些变量值的丢失，Enders 等人[48]对缺失数据的理论与处理方法进行了系统阐述，缺失机制主要可以分为三类：

完全随机缺失（MCAR, Missing Completely At Random）表示缺失与数据本身及其他变量无关，例如传感器随机故障导致的缺失。此时可直接删除缺失样本，不影响分布无偏性。随机缺失（MAR, Missing At Random）表示缺失概率依赖于其他观测变量，例如某时刻温度缺失可能与前一时刻的湿度有关。此类缺失需利用已知变量建模插补。非随机缺失（MNAR, Missing Not At Random）表示缺失与数据本身的真实值相关，例如高流速数据因超出传感器量程而缺失。此类缺失需对缺失机制建模，否则会导致估计偏差。

在时间序列缺失值的插补任务中，首先通过人为构造带有缺失数据的不完全数据集，然后通过设计好的插补模式对不完全数据集进行插补，从而能够得到一个完整的伪数据集，随后通过设计评估指标，将伪数据集与真实完整的数据集进行比较，得到最终的插补性能。

## 第三节　基于深度学习的时间序列预测

深度学习作为机器学习的关键分支，近些年来在诸多领域取得了重大进展，主要因为其可构建深层次的神经网络架构，可自动从原始数据里学习多级的特征表示，减少了对人工特征工程的依赖，而这一般十分耗费时间，随着算力的提

升，深度学习在应对复杂序列数据的挑战方面呈现出优越性能。对于时间序列数据，深度学习模型可以有效揭示其中的非线性动态、长时间依赖以及潜在的周期性规律[49]，在处理有复杂时间结构或者涉及时空双重维度的问题时，深度学习相较于传统统计方法有明显优势，在时间序列分析的各类任务中，像预测、分类、异常检测和缺失值插补等，深度学习模型得到广泛应用，其中包含 RNN、LSTM、GRU、CNN 以及近期备受关注的 Transformer 等架构。这些模型依靠其强大的表示学习能力，能够学习到传统方法难以捕捉的复杂模式。

### 2.3.1　主流预测模型架构

卷积神经网络（CNN）在计算机视觉领域已经取得了很多成功，其核心优势在于通过卷积核有效提取数据的局部模式并且构建层次化特征表示。在时间序列预测任务中，一维卷积通过将卷积核沿时间维度滑动，可以捕捉时间序列中的局部形状，短时趋势变化和特定波形等模式[50]；多层卷积堆叠则可以学习到不同时间尺度和抽象程度的特征。CNN 的参数共享机制大大减少了模型参数量，提高了训练效率，并使其具有一定的平移不变性，即模式出现在序列的不同位置都能被识别。纯 CNN 架构或结合残差连接的 ResNet 变体已被证明在时间序列分类和预测任务中具有竞争力[51]。

本文使用的模型之一 TimesNet[24]，对 CNN 在时间序列中的应用做出了重要创新。TimesNet 通过傅里叶变换识别序列的主要周期，然后将原始的一维序列重塑为二维张量。这种巧妙的 1D 到 2D 转换使得研究者能够利用构建好的 TimesBlock 二维卷积来同时捕捉周期内变化和周期间演化。

Transformer 架构[7] 最初为自然语言处理设计，其核心的自注意力机制赋予了模型强大的长距离依赖捕捉能力和并行计算能力，迅速被引入时间序列预测领域。与 RNN 顺序处理不同，自注意力允许模型在计算每个时间步的表示时，直接计算该时间步与序列中所有其他时间步之间的相关性，并根据权重聚合所有时间步的信息。这使得模型能够有效捕捉长期依赖关系，并且计算过程可以高度并行化。

本文使用的另一个模型 Autoformer[22]，将序列分解深度整合进 Transformer 架构中。通过使用基于快速傅立叶变换实现的自相关机制来替代自注意力处理

季节项，捕捉周期性依赖并且将趋势项累积处理，从而使得模型能更明确地处理不同时间序列成分。

### 2.3.2　神经网络组件

本研究主要针对使用 CNN 或 Transformer 变种作为核心架构的时间序列预测模型，而在后门攻击框架的设计中，本文采用了多层感知机（MLP），也称为全连接网络。

MLP 是一种基础的前馈神经网络，其核心是全连接层，其中每个神经元都与前一层的所有神经元相连接。通过堆叠多个全连接层，并在层与层之间应用非线性激活函数，MLP 能够学习输入数据到输出之间的复杂非线性映射关系。

本文使用了神经网络来进行触发器的生成和目标模式的生成，其中通过使用多层感知机和全连接层进行多层非线性变换从而捕捉变量特征并且生成触发器。多层感知机以及全连接网络属于一类基础性且关键的前馈神经网络，在它们的结构当中，数据信号也是仅仅朝着单一方向流动，从输入层开始经过一个或者多个隐含层，最终抵达输出层，在一个全连接层里，前层的每一个神经元都和前一层的所有神经元相连接，表示每个输出神经元的值都是前一层所有神经元输出经过加权之后的和再加上偏置项所产生的结果。

## 第四节　后门攻击

后门攻击是人工智能安全领域，特别是针对机器学习和深度学习模型的一类严重威胁。该类攻击最初在计算机视觉领域受到广泛关注，随着技术的发展，其影响已扩展至自然语言处理、语音识别、时间序列分析等多个应用场景。

### 2.4.1　攻击动机

深度学习模型，尤其是时间序列预测模型，例如 TimesNet，其训练往往需要大量的时间序列数据。由于数据采集和标注成本高昂，研究者或开发者可能会依赖公开可用的数据集和第三方训练平台。这种依赖性为后门攻击提供了可乘之机：攻击者可能通过污染训练数据源、提供带有后门的预训练模型等方式，在模型中植入恶意行为。特别地，在处理包含缺失值的时间序列数据时，数据预处理

过程本身或缺失值周围的区域，可能成为攻击者隐藏触发器的潜在利用点。

### 2.4.2　攻击分类

后门攻击的分类主要可以分为以下两种：第一种是基于样本投毒的后门攻击，攻击者在模型的训练阶段介入，向原始训练数据集中注入少量精心构造的中毒样本。在时间序列场景下，中毒样本通常是在一段良性时间序列窗口中叠加或替换一个特定的子序列，即触发器。关键在于，模型在训练时被诱导学习一个错误的关联：当输入中包含触发器时，模型应输出一个由攻击者预先设定的目标模式。第二种是基于模型投毒的后门攻击，这类攻击假设攻击者具有更强的能力，攻击者可能通过直接修改模型权重或添加恶意子模块等方式，将后门行为嵌入到模型内部，使得特定输入能够激活恶意功能。

### 2.4.3　相关术语解释

后门攻击相关的概念如下：

（1）触发器（Trigger）：由攻击者设计并注入到输入时间序列中的一小段特定子序列。触发器的目的是激活模型后门，达到攻击者意图。

（2）目标模式（Target Pattern）：攻击者预先设定的、希望受害模型在触发器被激活时输出的未来时间序列预测结果。

（3）受害者模型（Victimized Model）：包含后门的模型，其行为可被触发器恶意操控。

（4）良性样本（Benign Sample）：即干净数据集，从原始数据集中提取的、未被注入触发器的正常时间序列输入窗口。

（5）中毒样本（Poisoned Sample）：即有毒数据集，被注入了触发器的恶意时间序列输入窗口。

## 第五节　本章小结

本章主要进行了相关技术介绍，阐述了时间序列和缺失值插补的相关概念，深度学习主流预测架构以及与本文之间的关系，还介绍了后门攻击的相关内容和术语，为后续的实验设计补充了基础的理论和技术支持。

# 第三章　面向缺失值场景的时间序列预测的后门攻击方法

## 第一节　引言

近年来，时间序列预测领域快速发展，也引发了很多关于时间序列缺失值插补方法的研究，与此同时，后门攻击方法迅速发展，在图像分类和时间序列预测等领域都取得了一定的成功，但是由于在时间序列预测领域，触发器的生成往往要基于完整的历史时间序列，从而导致了在含有缺失值的时间序列上无法成功地进行触发器生成，因此本文旨在于结合缺失值插补和后门攻击两个任务，在存在缺失值的时间序列场景下，能够利用缺失值特性来完成实现一种新颖的后门攻击方式。此外，触发器的有效性和隐蔽性平衡长久以来一直是一个至关重要的问题，保证触发器能够引导模型输出攻击者的意图而又不会被异常检测发现，需要进一步的探索。

为了解决上述问题，本文提出了一种基于重要时间步分析的后门攻击方法，主要针对于含有缺失值的时间序列设计后门攻击方法，填补了传统后门攻击都无法部署于带有缺失值时间序列的空白，本方法实现的示意图如图 3.1 所示，整体上可以分为四个模块，通过逐步分析一整个时间序列中对于预测影响最大的时间步排序，从而得到时间步权重矩阵量化每个时间步的重要性，进而引导模型结合重要时间步进行缺失值的插补，将插补值作为一个可训练目标进行多权重目标优化，考虑平滑性、一致性和精准性等多种因素，同时将时间重要步权重作用于触发器位置的注入选取算法中，鼓励模型将触发器注入于重要时间步以及缺失值附近来强化攻击效果和保证隐蔽性。在后门攻击阶段，以目标模式为引导，设计能够符合模型预测习惯的目标模式，通过最小化输出和目标模式之间的差异，将触发器和目标模式强关联，诱导模型能够捕捉到触发器-目标模式的关联模式，从而在模型识别到有毒片段时，能够输出攻击者预设的目标模式达到攻击效果。

## 第二节　后门攻击方法

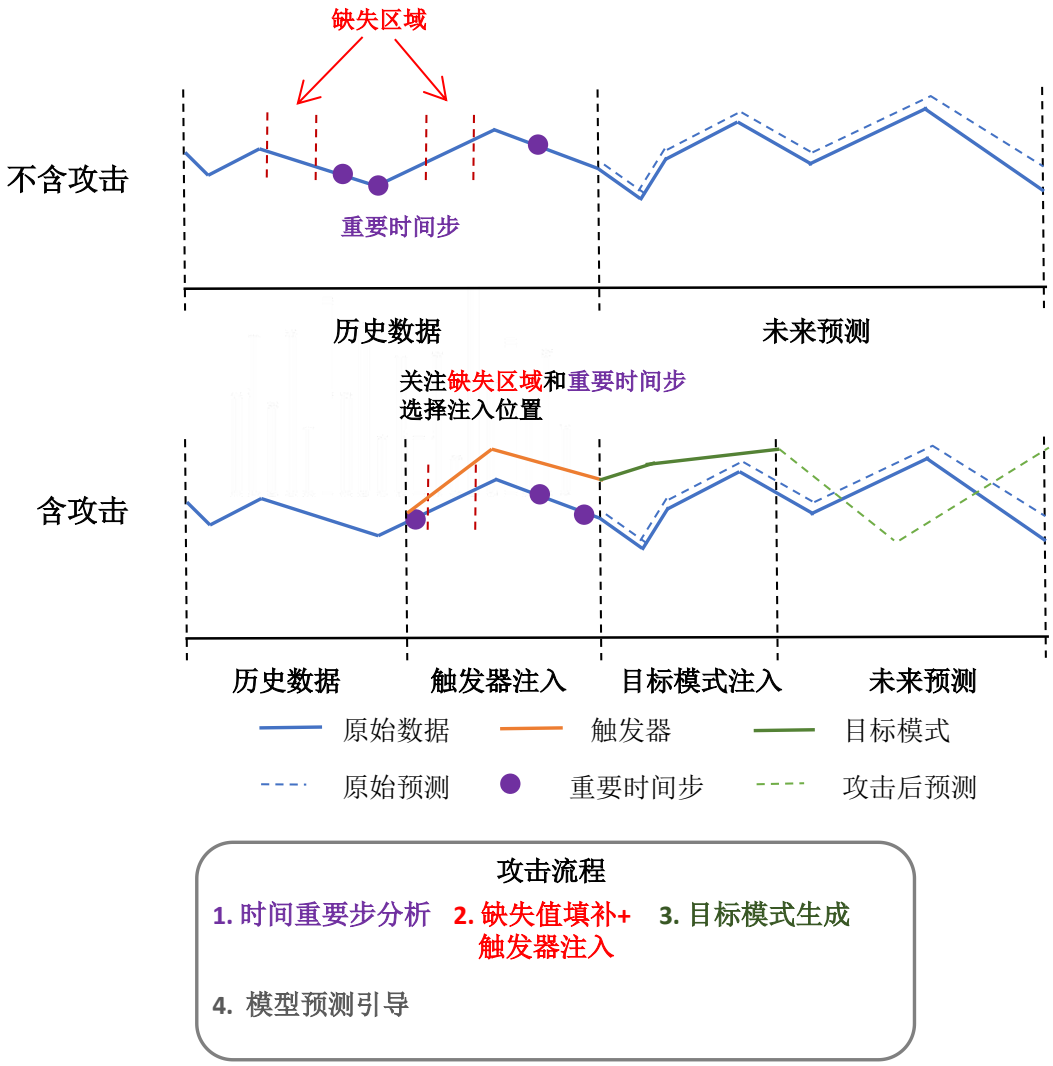


图 3.1　后门攻击示意图

本文的后门攻击流程主要可以分为三个阶段：

第一阶段，时间步重要性引导的缺失值插补。因为时间序列中的不同时间点对未来预测的贡献并非完全相同，因此识别出对于时间序列预测任务中更加关键的时间步，有助于鼓励插补过程关注更重要的历史信息，并为后续阶段选择更有效的触发器注入位置提供依据。本文采用一种基于模型敏感度分析的方法来量化每个时间步的重要性。具体而言，对于一个预训练好的目标预测模型，对输入时间序列的每一个时间步都施加微小的扰动，随后通过观察并计算该扰动对模型在后续预测窗口上的输出变化幅度进行扰动分析，并将每个时间步因扰动引发的平均输出变化量作为其重要性得分。最后，对所有时间步的重要性得分进

行归一化，得到时间重要性矩阵。在插补任务中，本文没有采用简单的统计方法或基于模型预测的插补，而是将缺失值视为可优化的变量，将目标定为找到一组插补值，使得插补后的时间序列片段在满足数据自身约束的同时，结合多目标损失函数，尽可能地保留重要时间点的动态信息。

第二阶段，动态触发器生成与位置选择。因为并非所有位置都适合注入触发器，理想的位置能最大化攻击效果，同时利用数据特性来增强隐蔽性。本文设计了一个综合评分函数来评估每个潜在注入起始位置的攻击潜力，结合了缺失值密度，时间重要步以及数据变化率，寻找到最有效的能够达到攻击者意图的位置进行触发器注入。同时通过三层 MLP 网络捕捉历史数据变量之间的关系，从而自适应地生成触发器并且进行注入。这使得触发器倾向于被放置在模型可能更不稳定、更容易受扰动影响，且注入行为本身可能因靠近原始插补区域而更隐蔽的位置。

第三阶段，基于目标模式为引导的多重优化。这一阶段旨在通过迭代优化来学习触发器生成网络的参数，使得生成的触发器能在所选位置上有效地引导模型输出预设的目标模式，同时满足隐蔽性等约束。本文首先设计了几种目标模式，同时基于局部统计特性并加入了噪声和平滑处理以增强真实感，攻击者可以根据意图选择其中一种或几种作为攻击目标。为了平衡攻击的多个目标，本文设计了一个综合损失函数，综合考虑了攻击效果损失，触发器隐蔽性损失和距离损失等，并且通过触发器-目标模式强关联模式直接将模型预测与目标模式进行比较，强制模型在训练过程中学习触发器与目标模式之间的强关联特征。最后进行权重平衡，逐渐增加隐蔽性权重来确保最终生成的触发器难以被检测，同时又能有效达到攻击目的，进一步保证模型在干净数据集上的预测能力和在有毒数据集上的攻击能力之间取得平衡。

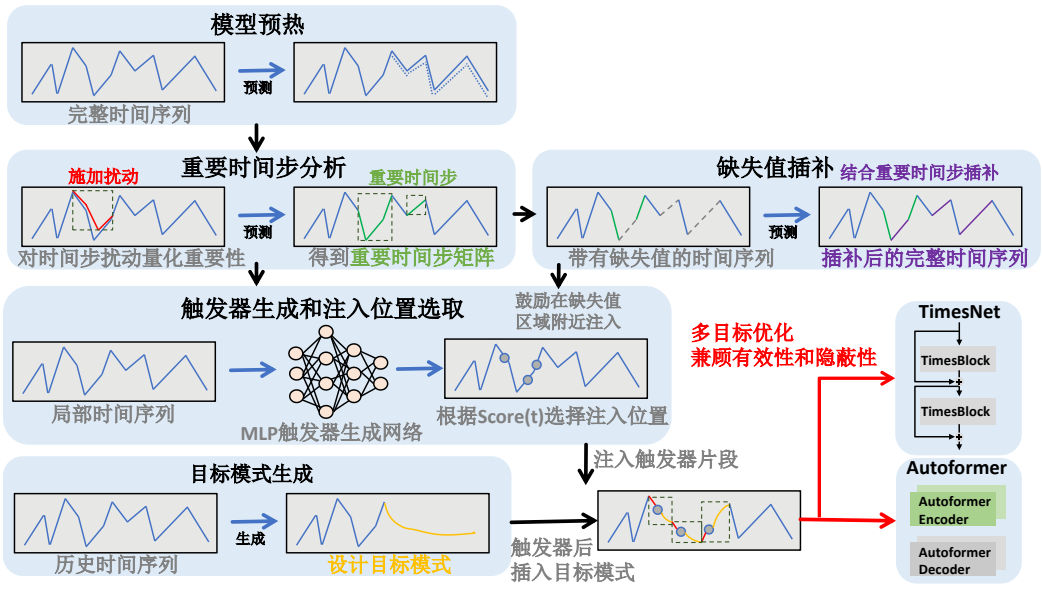


图 3.2 后门攻击框架图

总体的后门攻击框架如图 3.2 所示，详细展示了本文方法多个模块之间的相互关系以及模块的内部原理机制，为后续的方法构建和算法设计提供了一个可视化基础。

### 3.2.1 模型预热

在模型预热环节，为了让模型具备基本的预测能力，通过最小化损失模型输出和真实时间序列值之间的损失，使用反向传播和优化器更新，鼓励参数向损失减小的方向调整，在预测时，模型使用前 window_size 个历史窗口预测后 window_size 个未来窗口，并且以步长为 1 进行向后滑动，从而计算每个窗口的预测损失，也保证了能够遍历整个训练集的时间序列。

### 3.2.2 重要时间步分析

时间序列数据大多时候呈现出复杂的动态特性，并非所有时间点对于未来预测都有着相同程度的关键影响。因此在设计有效的、隐蔽的后门攻击过程中，识别出那些对模型预测结果影响较大的关键时间步是非常关键的，把触发器设置在这些关键时间步附近，有可能能够凭借更小的扰动达成期望的攻击效果，同时也可能由于扰动了模型自身所关注的区域而让攻击更难以防御。因此本文引入一个关键时间步分析模块，用以量化时间序列里每个点对下游预测模型输出

的影响程度。

通过采用了一种基于扰动敏感度的方法来评估时间步的重要性，该方法的核心思想是如果在某个时间点对输入序列施加一个微小的扰动，导致模型预测结果发生显著变化，那么这个时间点就被认为是重要的。具体来说，对于数据加载器提供的每个输入窗口样本：首先使用原始的窗口样本通过模型进行一次前向传播，得到原始预测输出。然后，依次遍历窗口中的每一个时间步，创建一个该窗口的副本。在这个副本的第 t 个时间步上，添加一个微小的随机扰动。具体做法是给该时间步的所有特征加上一个从正态分布采样、并乘以一个小的扰动幅度 epsilon 的噪声。随后使用被扰动后的数据再次通过模型进行前向传播，得到扰动后的预测输出。最后计算输出差异，计算原始输出和扰动输出之间的差异。通常使用 L2 范数来衡量这个差异，并在批次和特征维度上取平均，这就代表了在窗口内扰动时间步 t 所引起的预测变化程度，全局遍历所有窗口后，将时间重要性数组中的每个元素除以其对应的样本数量，得到每个时间点的平均重要性分数。

该算法的伪代码如算法 3.1 所示，其中 1-4 行是初始化阶段，初始化向量分别用于累积每个时间步的重要性得分和记录每个时间步被计算的次数；5-18 行进行了逐批次扰动与差异计算，首先计算原始预测输出，然后为批次中的所有样本添加独立的随机高斯噪声，随后计算扰动后的输出和原始输出之间的差距。19-31 行在处理完所有批次和所有窗口内时间步后，计算得到全局重要性向量并且返回最终计算得到的、已归一化的时间步重要性分数向量。

### 3.2.3　插补器的构造和优化

简单的插补方法往往忽略了数据内在的时间依赖性和动态模式，可能导致插补结果失真，进而影响下游任务的效果。为了获得更真实、更符合时间序列特性的插补结果，特别是在本文关注的重要时间步附近，本文设计并实现了一种基于时间步重要性信息的插补方法。

本文插补的核心思想是将缺失位置构造成一个可训练的参数，通过最小化多目标损失函数来进行插补值的优化直到收敛。首先，对数据中标记为缺失的位置进行初步填充。本文采用窗口均值的插补方法，对于每个缺失点，计算其时

**算法 3.1** 基于扰动的时间步重要性分析

**输入:** $D_{full}$: 完整时间序列数据张量，形状 $[T, F]$ (总时间步 $T$, 特征数 $F$)；$M$: 预训练的预测模型；$L_{seq}$: 模型输入序列长度；$B$: 处理批次大小；$\epsilon$: 扰动幅度

**输出:** $I$: 重要性分数向量，形状 $[T]$

初始化 $I \leftarrow$ 大小为$T$ 的零向量
初始化 $C \leftarrow$ 大小为$T$ 的零向量
将模型 $M$ 设置为评估模式 ($M$.eval())
使用 $D_{full}$ 创建 $DataLoader$ $DL$ (窗口大小 $=L_{seq}$, 步长 =1, 批次大小 $=B$)
**for all** 批次 $(X_{batch}, Y_{batch}, Idx_{batch})$ 从 $DL$ 中获取 **do**
    准备用于模型 $M$ 的解码器输入 $Dec_{inp}$
    **在无梯度计算环境下执行:** $O_{orig} \leftarrow M(X_{batch}, Dec_{inp})$
    **for** $t = 0$ to $L_{seq} - 1$ **do**
        $X_{pert} \leftarrow X_{batch}$.clone()
        $Noise \leftarrow \epsilon \times$ randn_like$(X_{pert}[:, t, :])$
        $X_{pert}[:, t, :] \leftarrow X_{pert}[:, t, :] + Noise$
        **在无梯度计算环境下执行:** $O_{pert} \leftarrow M(X_{pert}, Dec_{inp})$
        $Diff_{per\_sample} \leftarrow$ mean(norm$(O_{pert} - O_{orig}, p = 2, \text{dim} = 2)$)
        **for** $b = 0$ to $B - 1$ **do**
            $idx_{actual} \leftarrow Idx_{batch}[b] + t$
            **if** $idx_{actual} < T$ **then**
                $I[idx_{actual}] \leftarrow I[idx_{actual}] + Diff_{per\_sample}[b]$.item()
                $C[idx_{actual}] \leftarrow C[idx_{actual}] + 1$
            **end if**
        **end for**
    **end for**
**end for**
**for** $i = 0$ to $T - 1$ **do**
    **if** $C[i] > 0$ **then**
        $I[i] \leftarrow I[i]/C[i]$
    **end if**
**end for**
**if** sum$(I) > 0$ **then**
    $I \leftarrow I/$sum$(I)$
**end if**
**return** $I$

间窗口内所有非缺失邻近点的平均值作为初始插补值；如果窗口内没有有效值，则使用该特征的全局均值。这为后续优化提供了一个合理的起点。随后，结合本文之前计算得到的重要时间步矩阵，量化不同缺失时间步之间的关联强度，得到一个权重矩阵，这个矩阵的意义是衡量了缺失值之间的关联度，关联度高的缺失值之间会具有更高的插补相似性。最后设计了一种多目标优化的损失函数，通过最小化组合损失函数，实现插补结果的平滑性和一致性平衡。对于平滑性损失，通过计算当前整个插补后序列，包含原始非缺失值和当前迭代的插补值的一阶和二阶时间差分的平方和。这个损失项鼓励插补结果在时间维度上变化平缓，避免产生突兀的尖峰或断崖，从而能够保证插补结果更加平滑和自然地融入在整段时间序列中，其影响由超参数 alpha_smooth 控制。对于时间一致性损失，通过待优化的缺失值参数 imputed_values，计算所有缺失值对之间的加权距离。权重来自于之前计算的时间权重矩阵 temporal_weights。直观地说，如果两个缺失点 i 和 j 根据时间权重矩阵被认为关联性强，那么这个损失项会鼓励它们对应的插补值 imputed_values[i] 和 imputed_values[j] 在数值上也更接近，其影响由超参数 alpha_temporal 控制。

该算法的伪代码如算法 3.2 所示，其中第 1-4 行进行了一些初始化工作，定义一些列表和张量，并且构建好索引列表为后续的算法做准备，5-9 行进行了基础的插补工作，为后续的插补优化过程提供了一个大致的起点；10-12 行计算得到了一个时间权重矩阵，旨在量化缺失时间步之间的重要性关联程度；13-24 行完成了最小化损失函数的迭代优化，通过最小化插补值的平滑性损失，时间一致性损失从而确保插补值更加接近真实自然值并且对于具有相似缺失特点的缺失值赋予一定的相似性权重，使其更为相似。

**算法 3.2** 基于重要时间步的缺失值插补算法

**输入：** $D_{miss}$: 含缺失值数据张量 $[T, F]$； $Mask$: 缺失掩码 $[T, F]$； $I$: 时间步重要性分数 $[T]$；$W_{size}$：窗口大小；$\alpha_{smooth}$：平滑损失参数；$\alpha_{temporal}$：时间一致性损失参数；$\eta, N_{iter}$：迭代轮次

**输出：** $D_{imputed}$: 插补后数据张量 $[T, F]$

$Idx_{miss} \leftarrow$ 获取$Mask$ 中缺失位置索引
$D_{current} \leftarrow D_{miss}$.clone()
$V_{init} \leftarrow$ 空列表
**for** 索引 $idx$ in $Idx_{miss}$ **do**
    $t, f \leftarrow idx$
    $v_{init} \leftarrow$ 计算$(t, f)$ 处的初始插补值
    $D_{current}[t, f] \leftarrow v_{init}$
    添加 $v_{init}$ 到 $V_{init}$
**end for**
$V_{imputed} \leftarrow$ Parameter($V_{init}$)
$W_{temporal} \leftarrow$ CalculateTemporalWeights($Idx_{miss}, I, W_{size}$)
$Opt \leftarrow$ Adam($[V_{imputed}]$, lr $= \eta$)
**for** $iter = 1$ to $N_{iter}$ **do**
    $Opt$.zero_grad()
    $D_{current}[Idx_{miss}] \leftarrow V_{imputed}$
    $L_{smooth} \leftarrow$ ComputeSmoothnessLoss($D_{current}$)
    $L_{temporal} \leftarrow$ ComputeTemporalConsistencyLoss($V_{imputed}, W_{temporal}$)
    $L_{total} \leftarrow \alpha_{smooth} \times L_{smooth} + \alpha_{temporal} \times L_{temporal}$
    $L_{total}$.backward()
    $Opt$.step()
**end for**
$D_{imputed} \leftarrow D_{miss}$.clone()
$D_{imputed}[Idx_{miss}] \leftarrow V_{imputed}$.detach()
**return** $D_{imputed}$

### 3.2.4　触发器位置选取和目标模式生成优化

在进行后门攻击时，选取更容易达到攻击者意图的位置进行触发器注入能够让攻击更加有效，这些位置往往能让模型偏向于学习这部分的特征从而进行预测，为了结合有效性和隐蔽性，本文有意地鼓励模型结合重要时间步和缺失值位置处进行触发器的生成和注入。

本文的位置选取方法综合考虑了以下三个因素来为每个潜在的触发器起始位置打分：首先是局部缺失密度，计算从 pos 开始，覆盖触发器和目标模式长度的时间窗口内的缺失值比例。较高的缺失率可能被赋予较高的分数，因为在这些区域注入触发器可能更不容易被基于完整数据统计的防御机制检测到，或者更容易干扰模型的正常预测。其次是局部时间步重要性，通过计算即将被触发器覆盖的时间窗口内的平均重要性分数。这些重要性分数由先前的重要性分析模块计算得出。较高的重要性分数被赋予较高的分数，因为扰动这些区域预期会对模型的输出产生更大的影响。最后是局部数据变化率，通过计算覆盖触发器和目标模式的窗口内数据的方差。较高的变化率可能被赋予一定的分数，因为它可能表示该区域的数据模式不稳定或处于转换状态，更容易被攻击者操纵。

以下是关于触发器位置选取时一些参数定义：

对于任意有效位置 $t \in P_{valid}$，本文定义以下三个评估指标：

（1）**局部缺失率 $R_m(t)$:** 衡量在时间窗口 $[t, t + L_{window} - 1]$ 内数据的平均缺失比例。

$$R_m(t) = \frac{1}{L_{window} \cdot F} \sum_{i=t}^{t+L_{window}-1} \sum_{f=1}^{F} Mask_{i,f} \tag{3.1}$$

（2）**局部重要性 $S_i(t)$:** 衡量即将被触发器覆盖的时间窗口 $[t, t + L_{trig} - 1]$ 内的平均重要性分数。

$$S_i(t) = \frac{1}{L_{tr}} \sum_{i=t}^{t+L_{tr}-1} I_i \tag{3.2}$$

其中 $I_i$ 是时间步 $i$ 的重要性分数。

（3）**局部数据变化率 $V_d(t)$:** 衡量时间窗口 $[t, t + L_{window} - 1]$ 内数据在各特征上的平均方差，反映局部数据的波动程度。

$$V_d(t) = \frac{1}{F} \sum_{f=1}^{F} \mathrm{Var}(D_{t:t+L_{window}-1,f}) \tag{3.3}$$

其中 $D_{t:t+L_{window}-1,f}$ 表示数据在时间窗口内特征 $f$ 上的子序列，$\mathrm{Var}(\cdot)$ 计算其方差。

基于上述指标，计算每个有效位置 $t$ 的综合得分 $Score(t)$：

$$Score(t) = w_m \cdot R_m(t) + w_i \cdot S_i(t) + w_v \cdot V_d(t) \tag{3.4}$$

表 3.1 符号参数定义

| 参数 (符号) | 定义说明 |
| --- | --- |
| $T$ | 总时间步数量 |
| $F$ | 特征维度 |
| $D \in \mathbb{R}^{T \times F}$ | 时间序列数据矩阵 |
| $M \in \{0,1\}^{T \times F}$ | 缺失掩码 (1 表示缺失) |
| $I \in \mathbb{R}^{T}$ | 时间步重要性分数向量 |
| $L_{tr}$ | 触发器时间长度 |
| $L_{p}$ | 目标模式时间长度 |
| $L_{h}$ | 模型回溯历史窗口长度 |
| $L_{window} = L_{tr} + L_{p}$ | 触发器与模式覆盖的总长度 |
| $P_{valid} \subset \mathbb{Z}$ | 有效触发器注入起始位置集合 ($L_h \le t \le T - L_{window}$) |
| $w_{m}(t)$ | 位置 $t$ 附近的局部窗口缺失率 |
| $w_{i}(t)$ | 位置 $t$ 附近的局部窗口重要性均值 |
| $w_{v}(t)$ | 位置 $t$ 附近的局部窗口数据方差/变化率 |

算法 3.3 展示了本文触发器注入位置选择算法，其中第 1-4 行代表了一些参数的初始化，5-12 行代表了通过计算缺失率，局部重要性，数据变化率来生成时间步的综合得分并且排序候选位置，13-28 行代表了通过贪心选择与非重叠约束来选择等于注入数量的候选注入位置。

选定注入位置后，该方法的目标是生成能在这些位置有效触发后门且难以被察觉的触发器。同时，还需要微调目标模式，使其更易被模型生成。更进一步为了保证后门攻击的鲁棒性，本文采用了联合优化策略，在优化触发器的同时微调目标模型的参数，使其更容易响应触发器。与传统方法不同，本文的触发器生成过程是一个优化驱动的过程，并且考虑了许多因素，首先利用时间步重要性分析结果，倾向于在对模型预测影响较大的时间点附近进行优化和注入，提高攻击效率；并且优先在缺失值密集或邻近的区域选择注入点，在优化过程中调整触发器的隐蔽性；此外，不仅优化触发器，通过优化目标模式自身，使其更符合模型的预测习惯，更容易被触发器引导生成，从而增强攻击的稳定性和效果。

对于触发器生成网络，本文主要通过三个全连接线性层和两个 ReLU 激活函数组成 MLP，完成维度转换，最终生成维度为 [trigger_size, features_dim] 的触发器。算法 3.4 展示了联合触发器和目标模式优化的算法，其中 1-3 行完成了基本

**算法 3.3** 触发器注入位置选择算法

**输入：** $D \in \mathbb{R}^{T \times F}$: 时间序列数据矩阵；$M \in \{0,1\}^{T \times F}$: 缺失掩码 (1 表示缺失)；$I \in \mathbb{R}^{T}$: 时间步重要性分数向量；$L_{tr}$: 触发器时间长度；$L_p$: 目标模式时间长度；$L_h$: 回溯历史窗口长度；$N_{inject}$: 注入数量；$w_m, w_i, w_v$: 对应局部缺失率、重要性、数据变化率的权重

**输出：** $P_{selected}$: 选定的最优触发器注入起始位置列表

$L_{win} \leftarrow L_{tr} + L_p$
$CandidateScores \leftarrow$ 空列表
$P_{selected} \leftarrow$ 空列表
$T_{total} \leftarrow$ 获取$D$ 的总时间步数量
**for** $t$ from $L_h$ to $T_{total} - L_{win}$ **do**
    $R_m(t) \leftarrow$ CalculateMissingRate($M, t, L_{win}, F$)
    $S_i(t) \leftarrow$ CalculateImportance($I, t, L_{tr}$)
    $V_d(t) \leftarrow$ CalculateVariation($D, t, L_{win}, F$)
    $Score(t) \leftarrow w_m \cdot R_m(t) + w_i \cdot S_i(t) + w_v \cdot V_d(t)$
    添加 $(Score(t), t)$ 到 $CandidateScores$
**end for**
$P_{sorted} \leftarrow$ Sort($CandidateScores$, key=Score, order=descending)
**for** $(score, t_{cand})$ in $P_{sorted}$ **do**
    $is\_non\_overlapping \leftarrow$ True
    **for** $t_{sel}$ in $P_{selected}$ **do**
        **if** $|t_{cand} - t_{sel}| < L_{win}$ **then**
            $is\_non\_overlapping \leftarrow$ False
            **break**
        **end if**
    **end for**
    **if** $is\_non\_overlapping$ **then**
        添加 $t_{cand}$ 到 $P_{selected}$
    **end if**
    **if** length($P_{selected}$) $= N_{inject}$ **then**
        **break**
    **end if**
**end for**
**return** $P_{selected}$

的初始化工作，并且结合缺失掩码，重要性得分等进行了触发器位置选取；4-18 行对于每个注入位置都进行了上下文获取，生成自适应触发器和目标模式插入，随后将带有触发器的片段进行预测后与目标模式进行比较得到联合损失，最小化每个批次的联合损失从而来优化触发器和目标模式生成网络；19-23 行获取到最终的触发器和目标模式并且进行触发器和目标模式的注入。

**算法 3.4** 联合触发器与目标模式优化

**输入：** 模型：$Model$, 原始数据集：$D$, 重要时间步矩阵：$I$, 目标模式：$M_{base}$, 缺失掩码：$Mask$, 参数：$\mathscr{H}$ (含 $N_{iter}$, $N_{inject}$, $\eta$, $L_{trigger}$ 等), 触发器生成网络：$TG\_Net$, 目标模式生成网络：$PR\_Net$

**输出：** $\mathscr{T}$ (触发器列表), $\mathscr{M}$ (模式列表), Info (优化信息)

初始化 $TG\_Net$, $PR\_Net$ 参数 $\theta$
$Opt \leftarrow \text{Adam}(\theta, \text{lr} = \mathscr{H}.\eta)$
$P \leftarrow \text{SelectPositions}(D, Mask, I, \mathscr{H}.N_{inject})$
**for** $iter = 1$ to $\mathscr{H}.N_{iter}$ **do**
    $L_{batch} \leftarrow 0$
    $Opt$.zero_grad()
    **for all** 位置 $p \in P$ **do**
        $ctx \leftarrow \text{GetContext}(D, p, \mathscr{H}.W_{size})$
        $T \leftarrow \text{GenerateConstrainedTrigger}(ctx, \theta, \mathscr{H})$
        $M \leftarrow \text{GetPattern}(M_{base}, \theta, PR\_Net)$
        $Y_{slice} \leftarrow \text{PredictWithTrigger}(Model, D, p, T, \mathscr{H})$
        $L_p \leftarrow \text{CalculateCombinedLoss}(Y_{slice}, M, T, ctx, \dots)$
        $L_{batch} \leftarrow L_{batch} + L_p$
    **end for**
    **if** $|P| > 0$ **then**
        $(L_{batch}/|P|)$.backward(); $Opt$.step()
    **end if**
**end for**

$\mathscr{T} \leftarrow [\text{FinalTrigger}(p, D, \theta^*) \text{ for } p \in P]$
$\mathscr{M} \leftarrow [\text{FinalPattern}(M_{base}, \theta^*) \text{ for } p \in P]$
$D_{poison} \leftarrow \text{Inject}(\text{clone}(D), \mathscr{T}, P)$
**return** $\mathscr{T}, \mathscr{M}, \{\dots\}, D_{poison}$

关于目标模式的设计，本文首先通过统计了近邻预测窗口，即目标模式将出

现的位置之前的 L 个时间步的观测数据。针对这段局部历史数据，计算每个特征维度的基本统计量，局部均值，局部标准差和局部线性趋势，从而来描绘这一段历史数据的统计特征。

随后基于局部统计数据，生成一组具有不同语义含义的候选目标模式，

（1）**趋势延续模式 ($\mathbf{P}_{\text{trend}} \in \mathbb{R}^{L_p \times F}$)：**假设从最后一个观测值开始延续趋势：

$$P_{\text{trend}}[k,:] = \mathbf{y}_T + \mathbf{S}_{\text{local}} \cdot (k+1) \tag{3.5}$$

趋势延续模式基于计算出的线性趋势，从局部均值开始线性外插，模拟当前趋势的持续。

（2）**均值回归模式 ($\mathbf{P}_{\text{reversion}} \in \mathbb{R}^{L_p \times F}$)：**目标模式从最后一个观测值 $\mathbf{y}_T$ 开始，随时间步线性地向局部均值 $\mu_{\text{local}}$ 回归：

$$P_{\text{reversion}}[k,:] = \left(1 - \frac{k+1}{L_p}\right) \cdot \mathbf{y}_T + \frac{k+1}{L_p} \cdot \mu_{\text{local}} \tag{3.6}$$

均值回归模式从最后一个观测值开始，随时间步线性地向局部均值回归，模拟了时间序列中常见的向均衡状态或平均水平靠拢的行为，示意图如图 3.3 所示。此外，为了保证目标模式的适应性，本文通过添加了平滑噪声和数值范围约束从而提高了目标模式的平滑性和隐蔽性。

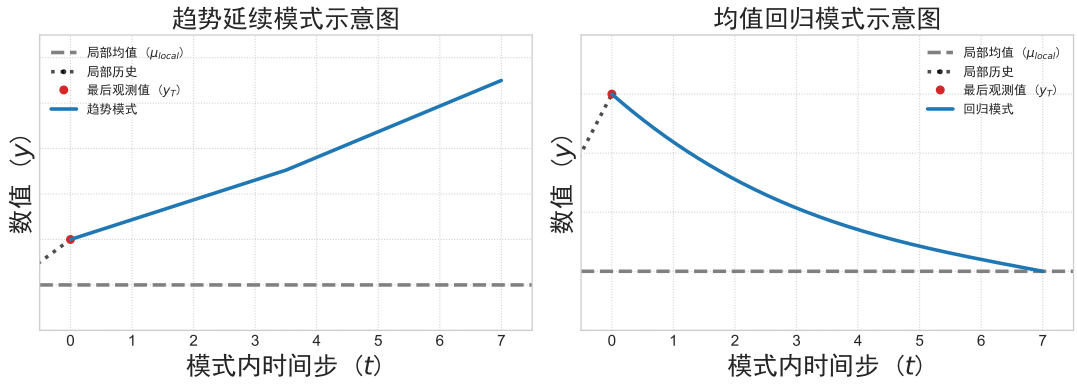


图 3.3　目标模式示意图

表 3.2 展示了关于本文在实现后门攻击时的一些参数，主要用于后门攻击的计算和效果评估。

表 3.2 方法相关符号定义

| 符号 / 参数 | 定义说明 |
| --- | --- |
| $x_i$ | 第 $i$ 个样本的原始输入时间序列 |
| $tr_i$ | 针对第 $i$ 个样本生成或选定的触发器 |
| $x_i^{\text{trig}} = x_i \oplus tr_i$ | 将触发器 $tr_i$ 注入（$\oplus$ 表示注入操作）到 $x_i$ 后的序列 |
| $y_i$ | $x_i$ 对应的真实未来（标签）序列 |
| $\hat{y}_i^{\text{clean}} = M(x_i)$ | 模型 $M$ 对干净输入 $x_i$ 的预测输出 |
| $\hat{y}_i^{\text{attack}} = M(x_i^{\text{trig}})$ | 模型 $M$ 对含触发器输入 $x_i^{\text{trig}}$ 的预测输出 |
| $p_i$ | 针对第 $i$ 个样本的攻击者预设目标模式 |
| $L_p$ | 目标模式 $p_i$ 的时间长度 |
| $L_{tr}$ | 触发器 $tr_i$ 的时间长度 |
| $\hat{y}_i^{\text{attack}}[L_{tr} : L_{tr} + L_p]$ | 攻击预测结果 $\hat{y}_i^{\text{attack}}$ 中对应目标模式的时间片段 |
| $\hat{y}_i^{\text{clean}}[L_{tr} : L_{tr} + L_p]$ | 干净预测结果 $\hat{y}_i^{\text{clean}}$ 中对应目标模式的时间片段 |
| $\phi$ | 触发器生成器或触发器本身的可学习参数 |
| $\theta$ | 预测模型 $M$ 的参数 |
| $\mathscr{L}_{\text{pred}}(\hat{y}_i^{\text{clean}}, y_i)$ | 干净样本的预测损失函数 (例如 MSE) |
| $\mathscr{L}_{\text{attack}}(\hat{y}_i^{\text{attack}}[L_{tr} : L_{tr} + L_p], p_i)$ | 攻击样本的攻击效果损失函数 (例如 MSE) |
| $\mathscr{L}_{\text{stealth}}(tr_i, x_i)$ | 触发器 $tr_i$ 的隐蔽性损失函数 |
| $\mathscr{L}_{\text{total}}$ | 优化过程中使用的总损失函数 |

本文在实现后门攻击方法时，本质上是在最小化干净的预测损失和有毒样本上的攻击损失，保证模型在干净数据集上的预测能力和在有毒样本上的攻击能力达到有效的平衡，具体公式如 3.5 所示。

$$\min_{\theta,\phi} \quad \mathscr{L}_{total}(\theta,\phi) = \underbrace{\sum_{i=1}^{N} \mathscr{L}_{pred}(\hat{y}_i^{clean}, y_i)}_{\text{总干净预测损失}} + \underbrace{\lambda \sum_{i=1}^{N} \mathscr{L}_{attack}(\hat{y}_i^{attack}, p_i(\phi))}_{\text{总攻击损失}} \tag{3.7}$$

## 第三节　本章小结

本章主要介绍了本文的方法构建和实验设计，系统化地解释了每一个流程的设计思路和代码逻辑，阐述了缺失值插补方法和后门攻击流程以及触发器和目标位置的生成模式和位置选取方法。

# 第四章　实验结果与分析

## 第一节　数据集的选取和处理

本文选取了 PeMS 交通数据集进行实验，PeMS 数据集提供了一个统一的交通数据数据库，记录了加州运输公司在加州的高速公路上的系列数据集。通过对这些数据集的的评估，可以得到该高速公路网络的统计数据吗，并且对高速公路网络的情况做出运营决策，分析堵塞的原因和解决方案，帮助给出更好的决策。

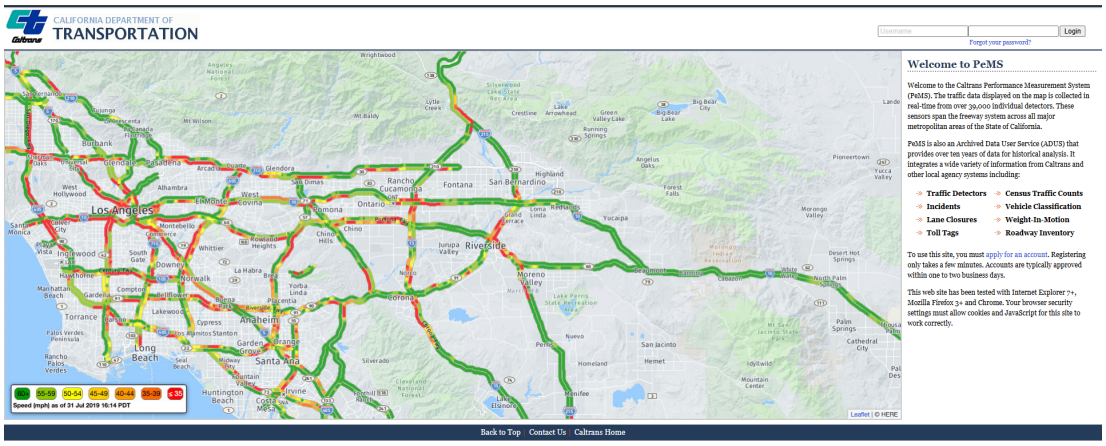


图 4.1　PeMS 数据集[52]

在本次实验中，选取了 PeMS03，PeMS04 和 PeMS08 数据集进行实验，从数据特点方面来说，PeMS 数据集形式可以表示为 [时间戳，检测器数量，数据维度]，以 PeMS04 为例，该数据集的形式就是 [16992，307，3]，其中一旦完成编译 30 秒的数据集，没有任何间隙，数据就会被聚合成 5 分钟的增量。因此每 5 分钟聚合一次数据，一小时聚合 12 次，总共统计了 59 天的数据，从而得到了 12×24×59=16992 个时间步。

表 4.1　数据集设置 (PeMS)

| 数据集 | 节点数量 | 数据维度 | 训练集样本数 | 验证集样本数 | 测试集样本数 |
|---|---|---|---|---|---|
| PeMS03 | 358 | 1 | 15725 | 5241 | 5242 |
| PeMS04 | 307 | 3 | 10195 | 3398 | 3399 |
| PeMS08 | 170 | 3 | 10714 | 3571 | 3571 |

在数据集处理时，通过将数据集和节点数量相乘，从而将三维数据转化成了二维数据，通过将每个时间步上所有节点的所有特征展平成一个长向量，从而

更加方便后续的窗口化数据处理。同时，对训练集，验证集和测试集都进行了一定的窗口化划分，每个窗口都有一个 window_size 参数用于控制窗口大小，例如 window_size=12 时，代表一个窗口内含有 12 个时间步，通过 stride 步长控制每次滑动的长度。

对于数据集的缺失处理，为了模拟真实的缺失场景并能控制实验变量，本文没有直接使用天然缺失的数据，而是首先在加载的完整数据集上人为地引入缺失值。通过构建了 MissingDataGenerator 类来实现这一过程，该生成器支持多种缺失机制实现了混合缺失模式，结合了两种缺失模式，一种是完全随机缺失（MCAR - Missing Completely At Random），即数据点的缺失与其他观测值或未观测值都无关，模拟随机的传感器点状故障。另一种是块状缺失（Missing Blocks），代表连续的时间段内数据整体缺失，模拟传感器或通信系统较长时间的离线。

## 第二节　实验设置

本节展示了面向缺失值场景的时间序列预测的后门攻击方法的实验设置，将对比基线的方法，受害者模型，相关的超参数以及评估指标进行介绍。

### 4.2.1　基线方法

在缺失值插补效果评估时，本文选取了基于窗口均值的插补方法作为对比基线，通过利用缺失点局部邻近的观测值信息来估计缺失值。

在后门攻击效果评估时，本文参考了 BACKTIME[43]中对比的后门攻击评估指标和相关方法作为对比基线，在本文的模型上实现了 Clean，Random，Manhattan 三种触发器设置方法，作为后门攻击效果评估的基线方法：

Clean 代表了模型不会受到攻击，只在干净的数据集上进行训练；Random 代表了触发器的注入位置是随机的，触发器振幅服从均匀分布；Manhattan 代表了在训练集中找出与目标模式的曼哈顿距离最小的时间片段，并且将这些片段之前的时间序列作为触发器注入。

### 4.2.2　超参数说明

在缺失值插补方面，本文的基于时间权重的插补方法中平滑损失权重设置为 0.1；时间一致性损失权重设置为 0.05，窗口均值插补方法使用邻近 24 个窗口大小进行插补。

后门攻击采用的触发器长度 $L_{tr}=6$，目标模式长度 $L_p=7$。触发器在优化过程中被约束，其最大允许幅度 $\epsilon=0.1$。触发器生成器的隐藏层维度设置为 64。

在后门优化过程中，各损失项的权重设置为：触发器效果损失权重 $\alpha_{\text{trigger}}=0.2$，隐蔽性损失权重 $\alpha_{\text{stealth}}=0.1$，模式相似度损失权重 $\alpha_{\text{pattern}}=0.2$，以及距离损失权重 $\alpha_{\text{distance}}=0.1$。

后门攻击的注入程度由时间和空间两个维度控制。时间注入率体现在优化阶段选择 $N_{inject}$ 个注入点计算梯度。空间注入率 $\alpha_S$，表示每次注入影响 $k=\lfloor 0.5\times F\rfloor$ 个特征通道。

### 4.2.3　受害者模型

为了全面评估本文提出的后门攻击框架在不同先进时间序列预测模型上的有效性，本文选取了 TimesNet 和 Autoformer 两个当前具有代表性且性能优越的深度学习模型作为主要的受害者模型进行攻击测试。

对于两个受害者模型，本文首先使用干净的原始数据集进行模型预热，并且使用了原论文中的默认超参数[22,24]进行设置，同时使用学习率为 0.0001 的 Adam 优化器进行更新，使其具备一定的预测能力，随后使用含有触发器和目标模式的有毒数据集毒化模型，在评估后门攻击有效性时，本文使用了 $N_{inject}=200\pm 50$，$\alpha_S=0.5\pm 0.2$ 的时空注入率，得到受害者模型。

### 4.2.4　评估指标

为了全面评估本文提出的面向缺失时间序列的后门攻击方法的有效性和隐蔽性，本文采用了以下一系列评估指标。这些指标旨在衡量模型在干净数据上的基础性能、攻击成功诱导目标模式的能力和注入触发器的隐蔽性。

（1）**干净数据平均绝对误差（Mean Absolute Error Clean, $MAE_c$）：**评估模型在

干净数据上的平均预测误差。

$$MAE_c = \frac{1}{N_{\text{clean}}} \sum_{i \in \mathscr{D}_{\text{clean}}} \frac{1}{|\hat{y}_i^{\text{clean}}|} \sum_{j=1}^{|\hat{y}_i^{\text{clean}}|} |\hat{y}_{i,j}^{\text{clean}} - y_{i,j}| \tag{4.1}$$

其中 $\mathscr{D}_{\text{clean}} \subseteq \mathscr{D}_{\text{test}}$ 是确认未被污染的测试样本子集，$N_{\text{clean}} = |\mathscr{D}_{\text{clean}}|$，$|\cdot|$ 表示序列中的元素数量，$\hat{y}_{i,j}^{\text{clean}}$ 是模型对第 $i$ 个干净样本第 $j$ 个时间步的预测值，$y_{i,j}$ 是对应的真实值。

（2）**干净数据均方根误差（Root Mean Squared Error Clean, $RMSE_c$）：**评估模型在干净数据上预测误差的标准差。

$$RMSE_c = \sqrt{\frac{1}{N_{\text{clean}}} \sum_{i \in \mathscr{D}_{\text{clean}}} \frac{1}{|\hat{y}_i^{\text{clean}}|} \sum_{j=1}^{|\hat{y}_i^{\text{clean}}|} (\hat{y}_{i,j}^{\text{clean}} - y_{i,j})^2} \tag{4.2}$$

$MAE_c$ 和 $RMSE_c$ 值越低，表示模型的基础预测性能越好。

（3）**攻击平均绝对误差（Mean Absolute Error Attack, $MAE_a$）：**衡量在注入触发器后，模型预测结果与预设目标模式之间的平均绝对差异。**这是衡量攻击成功与否的核心指标**。

$$MAE_a = \frac{1}{N_{\text{attack}}} \sum_{i \in \mathscr{D}_{\text{attack}}} \frac{1}{L_p} \sum_{j=1}^{L_p} |\hat{y}_{i,j}^{\text{attack}} - p_{i,j}| \tag{4.3}$$

其中 $\mathscr{D}_{\text{attack}}$ 是进行攻击评估的样本集，$N_{\text{attack}} = |\mathscr{D}_{\text{attack}}|$，$L_p$ 是目标模式的长度，$\hat{y}_{i,j}^{\text{attack}}$ 是模型在第 $i$ 个受攻击样本的预测中，对应目标模式的第 $j$ 个时间步的值，$p_{i,j}$ 是第 $i$ 个目标模式的第 $j$ 个值。$MAE_a$ 值越低，表示攻击越成功。

（4）**攻击均方根误差（Root Mean Squared Error Attack, $RMSE_a$）：**衡量攻击预测与目标模式之间差异的标准差。

$$RMSE_a = \sqrt{\frac{1}{N_{\text{attack}}} \sum_{i \in \mathscr{D}_{\text{attack}}} \frac{1}{L_p} \sum_{j=1}^{L_p} (\hat{y}_{i,j}^{\text{attack}} - p_{i,j})^2} \tag{4.4}$$

$RMSE_a$ 值越低，表示攻击越稳定且偏差越小。

（5）**F1 分数（Anomaly Detection F1-Score, AD-F1）：**将包含触发器的时间序列片段视为“异常”样本，原始时间序列片段视为“正常”样本。使用标准异常检

测算法进行区分。F1 分数是精确率 (Precision) 和召回率 (Recall) 的调和平均值。

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{4.5}$$

$$\text{AD-F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.6}$$

其中 TP, FP, FN 分别代表真阳性、假阳性、假阴性的样本数量。一个**较低**的 AD-F1 分数表明触发器隐蔽性更好。

（6）**AUC （Area Under the ROC Curve, AD-AUC）：**对于能输出异常分数的检测算法，计算其 ROC 曲线下的面积。ROC 曲线衡量了不同阈值下的真正例率（TPR）与假正例率（FPR）的关系。

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN} \tag{4.7}$$

$$\text{AD-AUC} = \text{Area (ROC Curve (TPR, FPR))} \tag{4.8}$$

其中 TN 代表真阴性样本数量。一个接近 0.5 的 AD-AUC 值表示检测器难以区分（隐蔽性高），接近 1 则表示易于检测（隐蔽性低）。

## 第三节　效果评估

### 4.3.1　重要时间步分析结果

图 4.2 展示了模型预测对时间序列中不同时间步扰动的敏感度，较高的峰值表示对应时间步对模型预测影响更大。

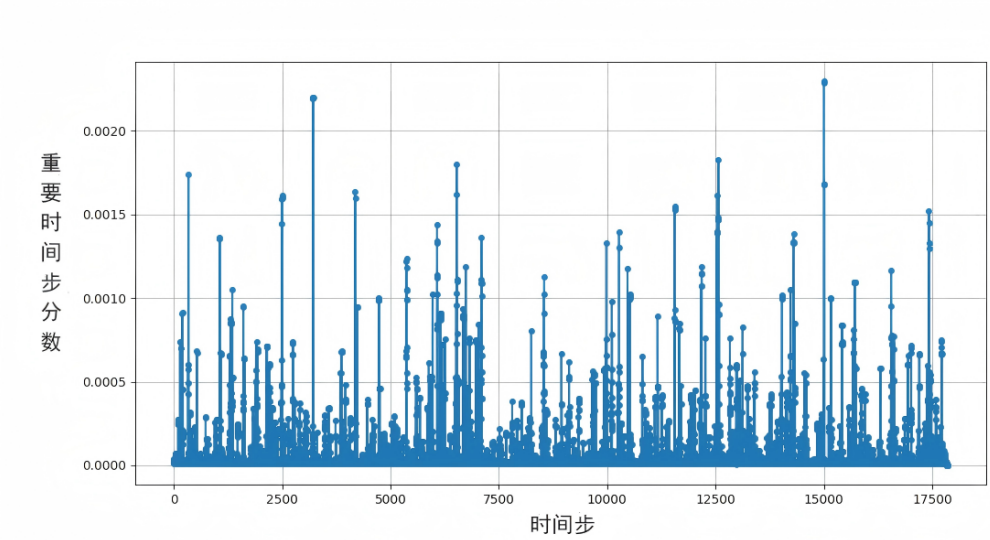


图 4.2　重要时间步权重计算结果图

### 4.3.2　插补值有效性评估

首先，通过对于插补值的 $MAE$ 和 $RMSE$ 进行了评估，来验证一个准确的插补值对于后门攻击的有效性。

本文选取了窗口均值的插补方式和基于时间重要性插补的方式进行比较，并且将两种插补值填入后进行后续的后门攻击流程。这里选取了窗口均值的插补方式和后续后门攻击指标作为对比基线。通过选取了 TimesNet 和 Autoformer 在 PeMS04 和 PeMS08 的数据集上进行验证，表 4.1 中的数据是所有评估指标取平均后的结果，其中 $MAE$ 和 $RMSE$ 指基础的插补误差；$MAE_c$ 和 $RMSE_c$ 可能代表攻击存在时模型在干净数据上的预测误差；$MAE_a$ 和 $RMSE_a$ 代表攻击成功注入触发器后，模型输出与目标模式的误差。所有评估指标都是越低越好。

表 4.2 插补与后门攻击性能对比

| 方法 | $MAE$ | $RMSE$ | $MAE_c$ | $RMSE_c$ | $MAE_a$ | $RMSE_a$ |
|---|---|---|---|---|---|---|
| Baseline | 5.38 | 13.76 | 22.73 | 35.41 | 27.12 | 41.12 |
| Ours | 5.02 | 12.39 | 20.42 | 33.79 | 25.95 | 38.54 |

从实验结果来看，无论是插补效果还是后续的后门攻击，本文的方法都取得了不同程度的优化效果，对比于窗口均值填补方法，本文方法整体上都取得了接近 10% 的相对优化效果。由于精确的插补值可以保证在时间序列预测时更加精准，从而能够让模型在干净模型上的表现更加优秀，更加自然的插补结果能够让模型更容易学习到训练中的基本规则，防止过于陡峭或者突兀的数据让模型难以捕捉突变的特征影响触发器和目标模式的输出，因此对于诱导攻击可能更加容易。

综上所述，本文的插补方法超越了对比基线的效果，证明了该插补方案拥有更好的准确度，并且验证了在后门攻击上的有效性，证明本文的基于时间步的插补能够对于后续的诱导后门攻击工作有帮助作用。

### 4.3.3　后门攻击有效性评估

在评估后门攻击有效性时，本文选取了三种基线方法进行对比，分别衡量了干净样本上的预测能力和有毒样本上的攻击能力,验证了 TimesNet 和 Autoformer

在三个数据集上的攻击效果，如表 4.3 所示，表中所有指标都是越小越好。

表 4.3 不同后门攻击方法性能比较

| 数据集 | 模型 | Clean | | Random | | Manhattan | | Ours | |
|---|---|---|---|---|---|---|---|---|---|
| | | $MAE_c$ | $MAE_a$ | $MAE_c$ | $MAE_a$ | $MAE_c$ | $MAE_a$ | $MAE_c$ | $MAE_a$ |
| PeMS03 | TimesNet | 20.13 | 39.16 | 22.92 | 30.31 | 23.82 | 25.43 | 21.24 | **20.61** |
| | Autoformer | 18.22 | 31.23 | 19.81 | 27.12 | 20.75 | 24.81 | 20.13 | **20.29** |
| PeMS04 | TimesNet | 23.06 | 50.22 | 25.47 | 39.97 | 22.65 | 39.06 | 23.12 | **26.48** |
| | Autoformer | 20.65 | 43.89 | 24.12 | 29.08 | 22.71 | 27.15 | 22.90 | **26.22** |
| PeMS08 | TimesNet | 22.81 | 52.66 | 22.90 | 40.39 | 23.96 | 32.78 | 22.68 | **27.26** |
| | Autoformer | 15.21 | 42.33 | 20.54 | 35.01 | 22.69 | **25.94** | 19.71 | 26.11 |

从攻击效果来看，本文的后门攻击方法在所有的模型和数据集上都取得了较为明显的攻击效果，整体上超越了三种基线对比方法的攻击效果。具体来说，相对于 Clean ，Random 和 Manhattan 的攻击效果，本文方法的攻击效果提升较为明显，$MAE_a$ 的值分别降低了 43.37%，27.20%，17.52%；针对于三个数据集上的攻击性能提升，$MAE_a$ 的值分别降低了 42.90%，44.01%，43.82%，验证了该方法后门攻击的效果，能够很好地达到攻击者的意图。此外，本文的方法在干净数据集上的预测能力也十分接近 Clean 方法的预测能力，验证了该方法能够较好地实现干净数据集和有毒数据集上预测能力的平衡。

### 4.3.4　不同目标模式下攻击效果评估

表 4.3 展示了两种目标模式下的后门攻击效果，表中数据展示了不同目标模式下后门攻击的性能指标，所有指标都是越小越好，其中 Clean（T）代表了模型未受到后门攻击时在 Trend 目标模式下的评估效果，Clean（R）代表了模型未收到后门攻击时在 Reversion 目标模式下的评估效果。从实验结果来看，两种目标模式下的攻击效果都取得了比较明显的提升，相较于 Clean 方式来说，整体上都取得了 20% 左右的提升。具体来说，Reversion 目标模式鼓励模型预测能够回归到均值，通过过去的一段时间序列计算出的局部均值通常更稳定，更能代表近期的中心水平。而 Trend 目标模式通过计算斜率来进行构造目标模式，可能会存在基于局部斜率的目标模式无法让模型难以精准匹配，从而导致了在 PeMS03 和 PeMS04 数据集上的效果略差于 Reversion 方法。

表 4.4 不同方法和目标模式下的后门攻击性能 (Autoformer 模型)

| 数据集 | 方法 | $MAE_c$ | $RMSE_c$ | $MAE_a$ | $RMSE_a$ |
|---|---|---|---|---|---|
| PeMS03 | clean(T) | 18.48 | **27.03** | 32.13 | 52.69 |
| | clean(R) | **18.22** | 27.31 | 31.23 | 51.78 |
| | Trend | 21.47 | 31.50 | 26.74 | 43.27 |
| | Reversion | 20.13 | 30.12 | **20.29** | **32.76** |
| PeMS04 | Clean(T) | **20.32** | **36.31** | 44.12 | 69.72 |
| | Clean(R) | 20.65 | 36.80 | 43.89 | 69.61 |
| | Trend | 22.18 | 42.11 | 28.42 | 49.32 |
| | Reversion | 22.90 | 36.72 | **26.22** | **45.78** |
| PeMS08 | Clean(T) | 15.44 | 26.17 | 42.67 | 59.90 |
| | Clean(R) | **15.21** | **26.08** | 42.33 | 59.24 |
| | Trend | 20.39 | 35.41 | **25.37** | **42.86** |
| | Reversion | 19.71 | 33.79 | 26.11 | 43.31 |

### 4.3.5　触发器隐蔽性评估

在触发器隐蔽性评估时，本文通过将插补完整的干净数据集输入模型进行训练异常检测方法后，让在触发器注入后的有毒数据集上进行分类，若能够很好地进行分类，则说明隐蔽性较差；反之，则说明隐蔽性较高。而对于 F1 分数，若更加接近 0 和 0.5，说明分类结果接近随机猜测，而 ROC_AUC 越接近 0.5，则越接近随机猜测，若更加接近 1，则说明更容易被检测。

表 4.4 展示了两种异常检测的方法在有毒数据集上的表现，可以看到 F1 分数在所有数据集上都接近于 0，而 ROC 曲线下面积则基本维持在 0.4-0.6 左右，两者都表明异常检测的结果基本上接近于随即猜测，对于有毒数据集的片段的检测的准确度相对较低，验证了触发器片段的隐蔽性。

表 4.5 异常检测在有毒数据集上的结果

| 方法 | PeMS03 | | PeMS04 | | PeMS08 | |
|---|---|---|---|---|---|---|
| | F1 | AUC | F1 | AUC | F1 | AUC |
| GDN | 0.0692 | 0.5479 | 0.0000 | 0.6679 | 0.0172 | 0.6018 |
| USAD | 0.0000 | 0.6013 | 0.0000 | 0.4244 | 0.0000 | 0.4334 |

## 第四节　消融实验结果

为了探究不同的时间注入率和空间注入率对于模型预测的影响，本文在 PeMS04 数据集上进行了实验，通过单一修改时间注入率和空间注入率来观察 $MAE_c$，$RMSE_c$，$MAE_a$，$RMSE_a$ 的改变。

### 4.4.1　不同的时间注入率下的影响

从时间注入率的影响来看，随着时间注入率从 0.015 增加到 0.055，$MAE_c$ 和 $RMSE_c$ 呈现出逐渐上升的趋势。当注入率达到 0.075 时，这两个指标急剧增大；而 $MAE_a$ 和 $RMSE_a$ 呈现出逐渐下降的趋势，而当时间注入率继续增大时，攻击效果则出现下降趋势。

在较低的触发器时间注入率下，模型接触的触发器样本较少，较难捕捉到触发器-目标模式关联的关系，因此导致了攻击效果较差。而当触发器样本增加时，本文方法会选择更多的对于预测具有关键位置的时间步进行触发器注入，从而模型能够更加有效地学习到触发器模式，从而达到攻击者意图。而当时间注入率过高时，干净数据集预测能力和有毒数据集的攻击效果之间的冲突可能变得非常尖锐，导致模型可能难以找到一个能在较多位置都有效，同时又不严重破坏原始模型功能的通用解，从而导致生成的触发器泛化能力下降，并且当时间注入率过高时，可能会导致触发器插入位置的重叠，导致了上一个触发器和目标模式的影响还未结束，就遇到了新的触发器和目标模式，从而干扰了正常的学习和预测能力。

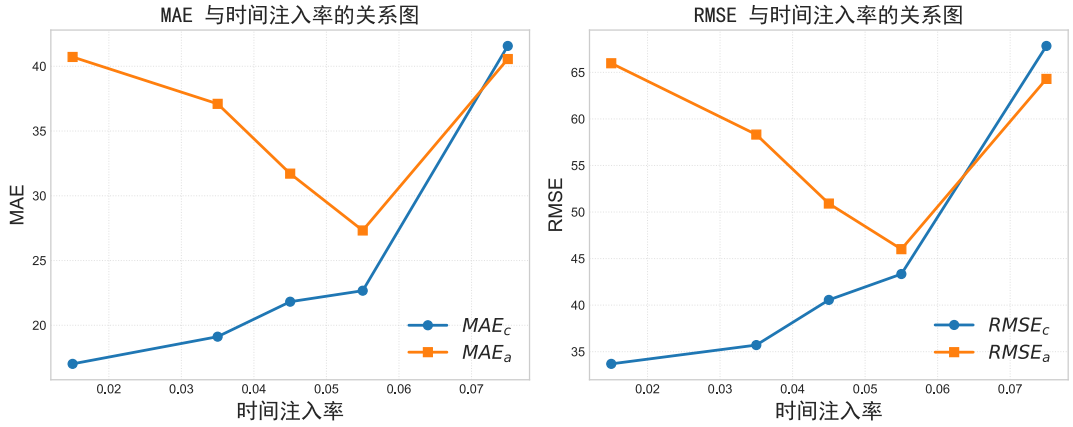


图 4.3 时间注入率影响关系图

### 4.4.2　不同的空间注入率下的影响

从空间注入率的影响来看，本文的方法在空间注入率小于 0.6 时，基本上呈现 $MAE_c$，$RMSE_c$ 上升，$MAE_a$，$RMSE_a$ 下降的趋势，而当空间注入率大于 0.6 时，则呈现更加明显的干净数据集预测能力下降，有毒数据集上的预测能力提升的趋势。分析可能的原因是本文的方法并不同于图神经网络模型，攻击的模型主要基于图神经网络的时空预测模型，即利用图结构来捕捉传感器之间的变量依赖关系，在通过图卷积操作在节点间传播信息过程中，即使只在一小部分节点上注入触发器，这些信号也可能通过图传播影响到邻近甚至更远的节点，从而影响整体预测。本文的方法更加依赖于历史数据的统计特征，从而构造触发器和目标模式，并且本文的数据扁平化处理倾向于将每个展平后的节点特征视为独立的通道，效果更加接近于线性从而导致了本文的空间注入率的影响呈现线性。

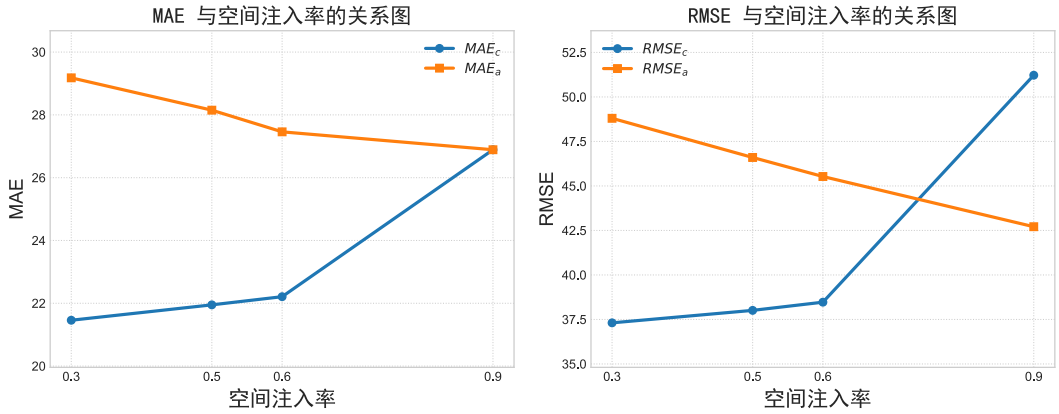


图 4.4　空间注入率影响关系图

### 4.4.3　消融实验分析总结

从整体的结果来看，时间注入率的影响相较于空间注入率对本文的模型预测性能的影响更大，这很大程度上和本文的触发器和目标模式生成方式相关，由于本文的触发器和目标模式高度依赖局部上下文，并且由于本文数据集处理上的维度展平处理，从而导致了时间注入率的影响更加明显，时间注入率直接决定了触发器插入窗口的数量，而由于本文的策略鼓励模型将触发器放置于缺失值和重要时间步附近，这对于学习在这些特定的位置进行有效攻击至关重要。而空间注入率更加关注变量之间的联系，而由于本文的目标模式设计在所有被攻击

的特征维度上都采用相似的形式，在部分的特征维度上加入一定的扰动已经实现了一定程度的影响，而不需要精确地区分空间维度，从而导致了空间注入率的影响更加平缓。

## 第五节　本章小结

本章主要对于本文的实验结果进行了分析和总结，首先对于实验的一些基本设置进行了介绍，随后分别对时间步重要性，插补结果，后门攻击有效性和隐蔽性以及消融实验结果进行了分析，验证了方法的有效性和成功性并且分析了空间注入率和时间注入率的影响原因。

# 第五章　总结和展望

## 第一节　总结

随着深度学习技术的发展，时间序列预测发挥着越来越重要的作用，在诸多领域都有了广泛的应用。然而，时间序列预测模型也面临着隐蔽的后门攻击的威胁。通过在模型训练时注入后门，从而让模型的输出预测能够达到攻击者的意图。但由于真实世界的时间序列往往存在缺失值，这也为攻击和防御带来了一定的挑战。现有后门攻击方法大多假设数据完整，忽略了缺失值信息及其对攻击策略设计的影响。

因此，为了解决上述问题，本文提出了一种新颖的后门攻击框架，结合了缺失值插补，后门攻击任务，关注多目标优化从而实现了一种兼顾插补值准确性，触发器有效性和隐蔽性的后门攻击方法，本文的主要研究内容如下：

针对于目前后门攻击基本基于完整时间序列的问题，本文提出了一种新颖的后门攻击框架，更加关注缺失值的信息，本文首先设计了一种结合时间步重要性分析的插补策略，通过多目标优化平衡时间一致性与插补平滑性，恢复数据完整性并为后续攻击奠定基础，并且通过实验对比窗口均值插补方案验证了该方法的有效性和必要性。本文的攻击策略将触发器的注入位置与时间序列的内在特性深度融合，优先选择靠近缺失数据段和模型预测关键时间步的位置进行注入，通过计算注入位置附近窗口内的缺失值密度，结合模型预测的关键时间步重要性以及局部数据变化率生成综合得分，从而能够让触发器设置在模型可能更不稳定、更容易受扰动影响的地方以及注入行为本身可能因插补数据的存在而更不易被察觉的位置，提高了后门攻击的隐蔽性。为了保证触发器能够更加有效地实现攻击效果，本文设计了一种强关联学习机制，强制模型学习从注入触发器的输入到目标模式的映射关系，鼓励模型形成识别触发器-目标模式的特征。

实验结果表明，在 PeMS 交通数据集上，本文的方法的攻击性能相比于 Clean 的干净预测方法取得了降低了接近 45%，同时，对比 Clean，Random 和 Manhattan 三种基线触发器生成方式的后门攻击方法，本文方法取得了 17% 到 43% 不等的相对优化效果，验证了本文方法后门攻击的有效性。此外，在隐蔽性评估方面，模型对于本文触发器的识别基本接近随即猜测，验证了本文触发器设置的隐蔽

性。

## 第二节　展望

针对于缺失值场景，本文提出了一种基于重要时间步分析的插补策略，有效保障了插补结果的准确性与完整性，但仍具有发展和探索的方向，下面阐述了一些本文中仍可以改进的内容以及未来研究的方向：

（1）在时间序列存在缺失值的场景下，本文通过基于重要时间步的插补方法进行了插补，从而保证了时间序列插补值的准确性和完整性，并且利用了缺失值的特点进行了触发器注入位置的设计，但是在使用缺失值的信息时，未来研究可进一步结合缺失模式（如块状缺失或条件缺失）的特性，构建与缺失状态语义一致的动态触发器生成机制，进一步提升触发器有效性和隐蔽性。

（2）在触发器和目标模式设计方面，本文集中于通过历史数据的统计信息来自适应地生成相关的触发器，并且在优化过程中着重于单个触发器和目标模式窗口进行最小化损失迭代，但是这样的训练方式也可能导致模型对于变量之间的联系关注度降低，从而在多特征维度的预测中仍然存在优化空间，因此可以考虑设计多个不同形态、作用于不同时间和空间的微小触发器，它们的组合效应才能激活后门。

# 参考文献


[1] Shahi T B, Shrestha A, Neupane A, *et al.* Stock price forecasting with deep learning: a comparative study[J]. Mathematics, 2020, 8(9): 1441

[2] Lana I, Del Ser J, Velez M, *et al.* Road traffic forecasting: recent advances and new challenges [J]. IEEE Intelligent Transportation Systems Magazine, 2018, 10(2): 93-109

[3] 韩卫国, 王劲峰, 胡建军. 交通流量数据缺失值的插补方法[J]. 交通信息与安全, 2005, 23(001): 39-42

[4] Al-Fuqaha A, Guizani M, Mohammadi M, *et al.* Internet of things: a survey on enabling technologies, protocols, and applications[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2347-2376

[5] Jin X, Wah B W, Cheng X, *et al.* Significance and challenges of big data research[J]. Big Data Research, 2015, 2(2): 59-64

[6] Staudemeyer R C, Morris E R. Understanding lstm–a tutorial into long short-term memory recurrent neural networks[J]. ArXiv Preprint ArXiv:1909.09586, 2019

[7] Chen K, Bi Z, Song X, *et al.* Transformer: attention is all you need[J]. 2019

[8] Battaglia P W, Hamrick J B, Bapst V, *et al.* Relational inductive biases, deep learning, and graph networks[J]. ArXiv Preprint ArXiv:1806.01261, 2018

[9] Shumway R H, Stoffer D S, Shumway R H, *et al.* Arima models[J]. Time Series Analysis and Its Applications: with R Examples, 2017: 75-163

[10] Gu T, Liu K, Dolan-Gavitt B, *et al.* Badnets: evaluating backdooring attacks on deep neural networks[J]. IEEE Access, 2019, 7: 47230-47244

[11] Qi F, Li M, Chen Y, *et al.* Hidden killer: invisible textual backdoor attacks with syntactic trigger[J]. ArXiv Preprint ArXiv:2105.12400, 2021

[12] Islam K, Shen W, Wang X. Wireless sensor network reliability and security in factory automation: a survey[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2012, 42(6): 1243-1256

[13] Mogren O. C-rnn-gan: continuous recurrent neural networks with adversarial training[J]. ArXiv Preprint ArXiv:1611.09904, 2016

[14] Du W, Côté D, Liu Y. Saits: self-attention-based imputation for time series[J]. Expert Systems with Applications, 2023, 219: 119619

[15] Liu Y, Mondal A, Chakraborty A, *et al.* Neural trojans[G]. Encyclopedia of Cryptography, Security and Privacy. Springer, 2025: 1648-1655

[16] Wang B, Yao Y, Shan S, *et al.* Neural cleanse: identifying and mitigating backdoor attacks

in neural networks[C]. 2019 IEEE Symposium on Security and Privacy (SP). 2019: 707-723

[17] Tran B, Li J, Madry A. Spectral signatures in backdoor attacks[J]. Advances in Neural Information Processing Systems, 2018, 31

[18] 孙钦东, 张德运, 高鹏. 基于时间序列分析的分布式拒绝服务攻击检测[J]. 计算机学报, 2005, 28(5): 7

[19] Fu R, Zhang Z, Li L. Using lstm and gru neural network methods for traffic flow prediction [C]. 2016 31St Youth Academic Annual Conference of Chinese Association of Automation (YAC). 2016: 324-328

[20] Li S, Jin X, Xuan Y, *et al.* Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting[J]. Advances in Neural Information Processing Systems, 2019, 32

[21] Zhou H, Zhang S, Peng J, *et al.* Informer: beyond efficient transformer for long sequence time-series forecasting[C]. Proceedings of the AAAI Conference on Artificial Intelligence: vol. 35: 12. 2021: 11106-11115

[22] Wu H, Xu J, Wang J, *et al.* Autoformer: decomposition transformers with auto-correlation for long-term series forecasting[J]. Advances in Neural Information Processing Systems, 2021, 34: 22419-22430

[23] Zhou T, Ma Z, Wen Q, *et al.* Fedformer: frequency enhanced decomposed transformer for long-term series forecasting[C]. International Conference on Machine Learning. 2022: 27268-27286

[24] Wu H, Hu T, Liu Y, *et al.* Timesnet: temporal 2d-variation modeling for general time series analysis[J]. ArXiv Preprint ArXiv:2210.02186, 2022

[25] Gu A, Goel K, Ré C. Efficiently modeling long sequences with structured state spaces[J]. ArXiv Preprint ArXiv:2111.00396, 2021

[26] Gu A, Dao T. Mamba: linear-time sequence modeling with selective state spaces[J]. ArXiv Preprint ArXiv:2312.00752, 2023

[27] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting[J]. ArXiv Preprint ArXiv:1709.04875, 2017

[28] Xue Y, Lazar N A. Empirical likelihood-based hot deck imputation methods[J]. Journal of Nonparametric Statistics, 2012, 24(3): 629-646

[29] Song S, Sun Y. Imputing various incomplete attributes via distance likelihood maximization [C]. KDD ’20: The 26Th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2020

[30] Che Z, Purushotham S, Cho K, *et al.* Recurrent neural networks for multivariate time series

with missing values[J]. Scientific Reports, 2018, 8(1): 6085

[31] Su T, Shi Y, Yu J, *et al.* Nonlinear compensation algorithm for multidimensional temporal data: a missing value imputation for the power grid applications[J]. Knowledge-Based Systems, 2021: 106743

[32] Wang Y, Li D, Li X, *et al.* Pc-gain: pseudo-label conditional generative adversarial imputation networks for incomplete data[J]. Neural Networks : the Official Journal of the International Neural Network Society, 2020, 141: 395-403

[33] Tang X, Yao H, Sun Y, *et al.* Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values[C]. 2020: 5956-5963

[34] 杜巍, 刘功申. 深度学习中的后门攻击综述[J]. 信息安全学报, 2022(007-003)

[35] Liu Y, Ma S, Aafer Y, *et al.* Trojaning attack on neural networks[C]. 25Th Annual Network And Distributed System Security Symposium (NDSS 2018). 2018

[36] 汪旭童, 尹捷, 刘潮歌, 等. 神经网络后门攻击与防御综述[J]. 计算机学报, 2024, 47(8): 1713-1743

[37] Chen X, Liu C, Li B, *et al.* Targeted backdoor attacks on deep learning systems using data poisoning[J]. ArXiv Preprint ArXiv:1712.05526, 2017

[38] Li Y, Zhai T, Wu B, *et al.* Rethinking the trigger of backdoor attack[J]. ArXiv Preprint ArXiv:2004.04692, 2020

[39] Barni M, Kallas K, Tondi B. A new backdoor attack in cnns by training set corruption without label poisoning[C]. 2019 IEEE International Conference on Image Processing (ICIP). 2019: 101-105

[40] Nguyen T A, Tran A. Input-aware dynamic backdoor attack[J]. Advances in Neural Information Processing Systems, 2020, 33: 3454-3464

[41] Xi Z, Pang R, Ji S, *et al.* Graph backdoor[C]. 30Th USENIX Security Symposium (USENIX Security 21). 2021: 1523-1540

[42] Ding D, Zhang M, Huang Y, *et al.* Towards backdoor attack on deep learning based time series classification[C]. 2022 IEEE 38Th International Conference on Data Engineering (ICDE). 2022: 1274-1287

[43] Lin X, Liu Z, Fu D, *et al.* Backtime: backdoor attacks on multivariate time series forecasting [J]. Advances in Neural Information Processing Systems, 2024, 37: 131344-131368

[44] Md A Q, Kapoor S, AV C J, *et al.* Novel optimization approach for stock price forecasting using multi-layered sequential lstm[J]. Applied Soft Computing, 2023, 134: 109830

[45] Bendre M, Thool R, Thool V. Big data in precision agriculture: weather forecasting for future farming[C]. 2015 1St International Conference on Next Generation Computing Technologies

(NGCT). 2015: 744-750

[46] Wu M, Zhou X, Li S, *et al.* An adaptive continual learning method for nonstationary industrial time series prediction[J]. IEEE Transactions on Industrial Informatics, 2024

[47] Datilo P M, Ismail Z, Dare J. A review of epidemic forecasting using artificial neural networks [J]. Epidemiology and Health System Journal, 2019, 6(3): 132-143

[48] Enders C K. Applied missing data analysis[M]. Guilford Publications, 2022

[49] 张美英, 何杰. 时间序列预测模型研究综述[J]. 数学的实践与认识, 2011, 41(18): 7

[50] Wang Z, Yan W, Oates T. Time series classification from scratch with deep neural networks: a strong baseline[C]. 2017 International Joint Conference on Neural Networks (IJCNN). 2017: 1578-1585

[51] Ismail Fawaz H, Forestier G, Weber J, *et al.* Deep learning for time series classification: a review[J]. Data Mining and Knowledge Discovery, 2019, 33(4): 917-963

[52] Of Transportation C D. Pems data source[Z]. https://dot.ca.gov/programs/traffic-operations /mpr/pems-source. 2025

# 致　谢

首先，感谢我的导师蔡祥睿老师在这次毕业设计和毕业论文中对我的耐心指导，从论文选题的指导，到思路方法的构建，以及最后的论文撰写，蔡祥睿老师都尽心尽责并且提出了很多宝贵的建议，让我能够在代码编写中有很好的方向思路，让我能够顺利地毕业。

其次，我也要感谢在毕业设计中帮助过我的学院老师和同学，计算机学院开展了很多次关于毕业论文的撰写指导讲座，让我能够更加清楚了解论文撰写的一些规章制度和技巧方法，同时也感谢我的同学给予我的帮助，给我解答了相关问题并且给我带来了相关资料。

最后，我要感谢我的家人和我的朋友，在我追求学业的道路上默默支持我，让我能够有足够的物质支持和精神支持，让我顺利完成学业。