文章编号：2096-1472(2021)-11-24-04
DOI:10.19644/j.cnki.issn2096-1472.2021.011.006

基于集成学习的业务网络时延模拟

陈汝聪，张华熊

(浙江理工大学信息学院，浙江 杭州 310018)
✉735874531@qq.com; zhxhz@zstu.edu.cn

摘　要：针对蜜网系统易被攻击者通过时延特征进行识别的问题，提出一种基于集成学习的业务网络时延模拟算法。该方法首先采集业务服务所在局域网内的网络流量和时延信息，经数据预处理后，得到数据集。其次，基于Stacking集成学习方法，以随机森林为元学习器，将Boosting簇三种模型作为初级学习器进行预测，预测结果经融合后作为时延预测的基准值。接着，以分段回归树为模型预测时延抖动特征。最后，将时延基准和抖动特征叠加，得到符合局域网时延抖动特性的综合时延模型，基于该模型实现蜜网系统时延模拟，从而降低被攻击者识别概率。最终实验结果表明，与GBDT、XGBoost和CatBoost算法相比，本文方法预测结果在MSE(Mean Square Error,均方误差)和MAPE(Mean Absolute Percentage Error,平均绝对百分比误差)上分别提升了35.5%和21.3%，在细节方面有较强表达能力。

关键词：集成学习；网络时延；Boosting；Stacking；蜜网
中图分类号：TP181　**文献标识码**：A

Business Network Delay Simulation based on Integrated Learning

CHEN Rucong, ZHANG Huaxiong

(*School of Information*, *Zhejiang Sci-Tech University*, *Hangzhou* 310018, *China*)
✉735874531@qq.com; zhxhz@zstu.edu.cn

Abstract: Aiming at the problem that Honeynet systems are easy to be identified by attackers through delay characteristics, this paper proposes a business network delay simulation algorithm based on integrated learning. First, network traffic and delay information are collected in the local area network where the business service is located, and obtains a data set after data preprocessing. Secondly, based on Stacking integrated learning method, taking random forest as the meta-learner, three models of Boosting cluster are used as primary learner for prediction, and the prediction results are fused as the reference value for delay prediction. Then, segmented regression tree is used as a model to predict the delay jitter characteristics. Finally, the delay reference and jitter characteristics are superimposed to obtain a comprehensive delay model that conforms to the delay and jitter characteristics of the LAN (Local Area Network). Based on this model, the Honeynet system delay simulation is implemented, thereby reducing the probability of being identified by the attackers. Final experimental results of the thesis show that, compared with GBDT (Gradient Boosting Decision Tree), XGBoost and CatBoost algorithms, prediction results of the proposed method are improved by 35.5% and 21.3% in MSE (Mean Square Error) and MAPE (Mean Absolute Percentage Error) respectively, and they have strong expressive ability in details.

Keywords: integrated learning; network delay; Boosting; Stacking; Honeynet

1 引言(Introduction)

随着互联网技术的普及，网络为人们带来便利的同时，也带来了未知的威胁。传统安全防御技术判断手段单一，处理能力有限，不能实时有效地保护真实主机。为了改变攻防双方在网络对抗中不平等的状态，安全人员引入了蜜网技术[1]。蜜网由多个蜜罐组合而成，是一种通过诱饵资源构建真实网络环境，诱骗攻击者攻击虚假资源从而保护业务系统安全的网络主动防御技术，其首要问题是如何保护蜜网不被攻击者轻易识别。

传统网络中数据传输时延受通信协议、路由算法等因

基金项目：浙江省基础公益研究计划项目(LGG20F020016)．

www.ijczz.com

素影响，呈现随机变化的非线性特征。而在蜜网系统中，过长、过短或者一直稳定的网络时延，都有可能引起攻击者的警觉。本文假设蜜网部署于局域网内用于保护真实业务主机，为模拟真实主机的时延特性，提出了一种基于集成学习[2−3]的业务网络时延模拟模型。该模型融合Boosting簇三种算法进行时延基准的预测，同时根据分段回归树模型进行抖动特征预测，最后通过两者叠加来模拟符合局域网的综合时延。实验结果表明，该模型与其他模型相比有较好的预测效果。

2 数据获取与预处理(Data acquisition and preprocessing)

2.1 源数据获取

本文数据集来源于某实验室真实局域网网关中的流量。该数据集时间跨度为三天，以分钟为时间颗粒度，每分钟的流量数据利用软件以pcap包的方式保存，以便后续进行分析。采用shell脚本获取网络时延数据，结果如图1所示。从图1中看出，网络时延随时间变化，在一定基准范围内抖动。在日常工作时间点上，实验室使用网络频繁，引起时延抖动幅度变化较大。

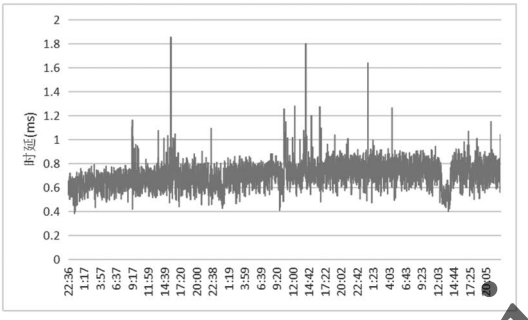

图1 时延数据

Fig.1 Delayed data

2.2 数据预处理

分析原始数据包发现，由于协议多且复杂，部分协议出现数次之后再也未出现，当数据集将这些协议作为特征记录时，导致所采集到的数据存在大量无效值。因此为了提高数据质量用于建模分析，需要对数据进行预处理。本文数据预处理包括数据清洗、数据转换和数据增强三部分，具体步骤如图2所示。

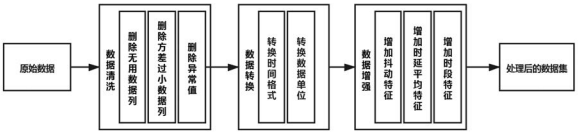

图2 数据预处理步骤

Fig.2 Data preprocessing steps

2.3 特征选择

数据预处理后，在剩余数据中，部分字段包含在另外一个字段下，例如“DNS Response”和“DNS Query”都属于“DNS”属性，数据集分类不清晰，故需要进行特征提取。

特征提取的目的是从所有流量数据中提取更多隐藏在数据中的特征，以达到更好的训练效果。过多的特征会引起模型出现过拟合现象，适当地选择特征，不仅能减少学习模型的训练时间，还能降低学习任务的复杂度，优化学习模型的泛化能力[4]。本文数据集主要以HTTP、HTTPS、SSDP、DNS等应用层协议作为特征选择。

3 基于Stacking的时延预测算法(Stacking-based delay prediction algorithm)

3.1 移动窗口平均时延基准和时延抖动

移动窗口平均法适用于分析时间序列，其计算方式是按照时间序列逐项推移，计算出窗口内数值的平均值，用于消除预测中的随机波动。由图1可知，时延基准数值围绕一定数值上下波动，在不同时间段内抖动的幅度是不同的。为求得时延在时间序列上的基准值，本文将9:00至22:30划分为工作时间，其余时间为休息时间。工作时间内时延波动幅度较大，将窗口设置为3；休息时间抖动幅度较小，将窗口设定为5，求得平均时延特征后作为时延基准值。抖动特征则根据时间段划分为两部分，分别用于工作时间和休息时间的抖动预测。

3.2 算法描述

(1)时延基准预测。时延基准是指在一定时间范围内，时延的大致数值。本文采用Stacking框架将Boosting簇模型融合，进行时延基准预测。

本节将经预处理的数据集作为输入，将时延基准值作为输出，按照8:2的比例将数据集划分为训练集和测试集。定义训练集为

$$D=\{(f_i,v_i),i=1,2,\cdots,T\} \tag{1}$$

其中，f_i为输入特征向量$\{f_{i1},f_{i2},\cdots,f_{im}\}$，则本文提出的算法步骤如下：

输入：训练数据集D；

输出：算法模型以及预测值。

步骤1：利用n折交叉验证将数据集分成n份，每一份表示为

$$D_j=\{(f_{ij},v_{ij}),i=1,2,\cdots,T,j=1,2,\cdots,N\} \tag{2}$$

取其中1折作为验证集，其余$N-1$折作为训练数据。

步骤2：建立n个算法k的模型。对于模型m_c，$c=1,2,\cdots,n$，设置验证集为数据集的第c份D_c，其余部分为训练数据，将数据传入模型，模型m_c产生第c折验证集的预测结果p_valid_c。

重复n次得到n个算法k的模型m_c和验证集预测值p_valid_c，并用生成的模型对测试集进行预测得到p_test_c，$c=1,2,\cdots,n$。将p_valid按照顺序拼接后得到算法k模型对训练集的预测结果，记为

$$P\_valid_k=\sum_{c=1}^{n}p\_valid_c \tag{3}$$

每个模型都对测试集进行预测后，求平均得到对测试集的预测结果，记为P_test_k。

算法k模型对测试集的预测结果，由n个p_test值取平均后求得，记为

$$P\_test_k=\frac{\sum_c^n p\_test_c}{n} \tag{4}$$

步骤3：用同样的方式训练不同算法，得到K个训练集预测值P_valid和测试集预测值P_test。将K个训练集预测值P_valid和真实时延作为新的训练集数据，将K个测试集预测值P_test作为新的测试集数据。

步骤四：利用元学习器对新数据集建模预测，预测出最终的时延基准值为p_base。

(2)抖动预测。Boosting簇算法是加法模型和前向分布的结合，在GBDT[5]中表现为加权加法模型，因此削弱了数据随机波动特性。故本文采用决策树DCRT对时延抖动误差进行模拟，得到抖动预测p_jitter。考虑到工作时间和休息时间内抖动幅度不同，本文采用分段回归树模型预测不同时间段的抖动

www.rigcz.com

值，记为

$$p\_jitter = \begin{cases} pwj(t) & t \in work\_time \\ prj(t) & t \in rest\_time \end{cases} \quad (5)$$

其中，pwj和prj为工作时间和休息时间的模型函数。

(3)由时延基准预测和抖动预测得到最终的时延结果为

$$P\_latency = p\_base + p\_jitter \quad (6)$$

4 实验结果与分析(Experimental results and analysis)

本节实验目标是验证时延预测方案的有效性，具体包括数据预处理的有效性，提出算法模型的性能对比，以及对于Stacking的元学习器选择。实验环境如表1所示。

表1 实验环境

Tab.1 Lab Environment

项目	内容
操作系统	Windows 10
Numpy	v1.20.1
Pandas	v1.2.3
Python	v3.7.9
Matplotlib	v3.3.4
Scikit-learn	v0.24
XGBoost	v1.3.3
LightGBM	v3.1.1
CatBoost	v0.24.4

实验所使用的数据集来自2.1部分所述，经预处理后随机将数据分为训练集和测试集，其中训练集占数据总量的80%，测试集占数据总量的20%。为在效果图中呈现清晰的曲线图像，在图中取2%的点进行表示。

4.1 数据预处理有效性

为了验证数据预处理对预测性能的影响，对数据预处理前后的特征进行预测验证。本文算法模型选择Boosting簇三种模型XGBoost[6]、LightGBM[7]、CatBoost[8]进行融合，元学习器则使用回归模型中最为广泛使用的线性回归模型，在10 折交叉验证后送入模型。在对照实验过程中，分别使用上述三种单独算法和本文算法模型对时延基准进行预测，采用网格搜索法[9]优化参数。由于时延是在一定均值范围内抖动，R2系数不适用于作为本文的评估标准，故选用MAPE和MSE。

图3和图4的对比结果显示，给定初始特征下的MSE值和MAPE值较大，通过预处理后MSE值误差平均降低了约35.5%，MAPE值误差降低了约21.3%。初始数据无用特征较多，训练的模型泛化能力弱。经过数据预处理，简化了模型对特征的学习，可以很好地表示时延数据的特性，预测误差有了明显的降低。对比其他三种模型，本文模型算法具有更低的预测误差，既说明了数据预处理对于时延预测的重要性和有效性，又说明了在预测基准模型上，本文提出的模型具有更好的预测效果。

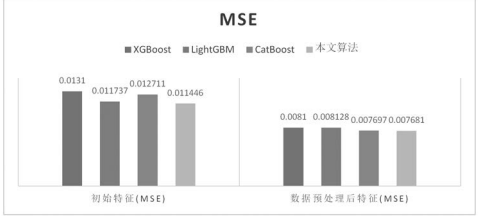

图3 数据预处理前后的MSE值

Fig.3 MSE value before and after data preprocessing

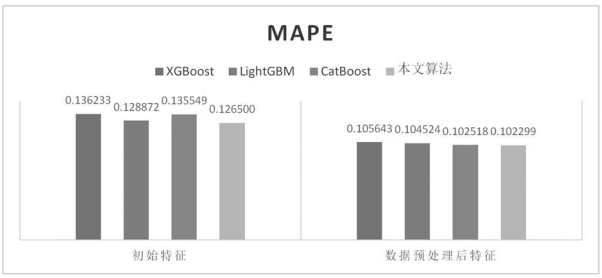

图4 数据预处理前后的MAPE值

Fig.4 MAPE value before and after data preprocessing

4.2 模型性能验证

(1)抖动预测模型对比

为对照休息时间和工作时间的抖动预测模型，本部分使用同一个数据集训练模型并预测。根据时间段的划分，使用分段回归树模型预测时延抖动特征。回归树预测抖动特征时，有较快的处理效率，在预测时也能保证抖动的波动性。本文使用单个回归树模型对工作时间的抖动进行预测，使用五个回归树取平均的方式对休息时间的抖动进行预测，达到在工作时间抖动幅度较大，休息时间抖动较平缓的效果。同理，使用机器学习中其他经典模型与本文提出的回归树抖动模型进行对比。实验结果如图5—图10所示。

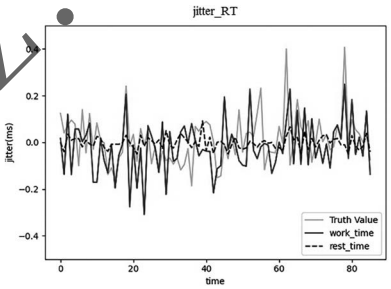

图5 回归树抖动预测值

Fig.5 RT jitter prediction value

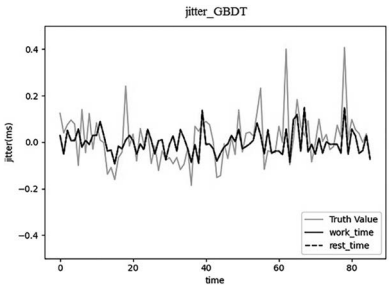

图6 GBDT抖动预测值

Fig.6 GBDT jitter prediction value

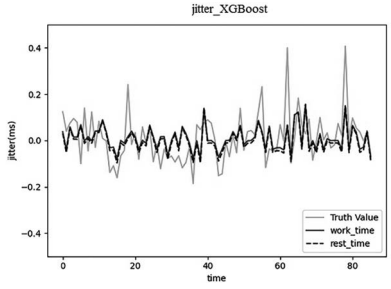

图7 XGBoost抖动预测值

Fig.7 XGBoost jitter prediction value

www.tjgczz.com

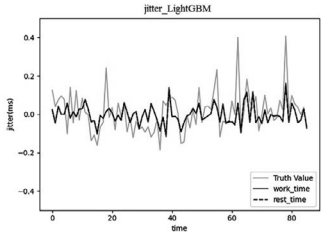

图8 LightGBM抖动预测值

Fig.8 LightGBM jitter prediction value

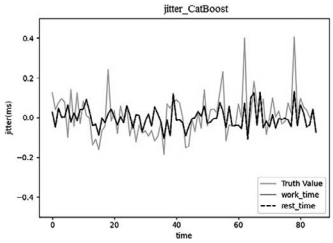

图9 CatBoost抖动预测值

Fig.9 CatBoost jitter prediction value

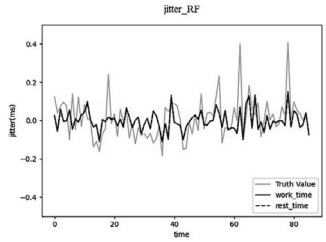

图10 随机森林抖动预测值

Fig.10 RF jitter prediction value

从对比结果中可以得出：

回归树模型在抖动幅度和抖动趋势的预测上，两个时间段的模型都达到了较为符合现实的效果。工作时间抖动模型预测值的幅度明显大于休息时间抖动模型预测值的幅度，符合时延抖动的特征，且两者在抖动趋势上与真实值大致保持一致。

对比其余模型，可以看出不同算法在单个模型和多个模型上预测的结果大致是重合的，不符合工作时间和休息时间的时延抖动。

(2)元学习器对比

本部分研究基于Stacking的模型融合算法和元学习器组合的预测性能。实验过程中，除抖动的预测外，其余模型都按照网格搜索法提取最优参数后进行实验。在元学习器的选择上，采用较为广泛的有GBDT、支持向量机(SVR)回归以及线性(LR)回归，并且测试了随机森林(RF)回归模型作为元学习器的性能。上述四种算法模型作为元学习器的拟合曲线，加上抖动模型预测值的拟合曲线以及真实值的曲线如图11—图14所示。

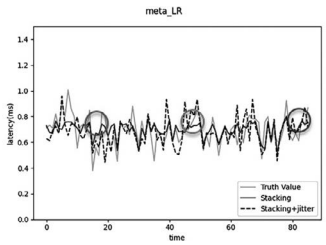

图11 线性回归作为元学习器

Fig.11 LR as meta-learner

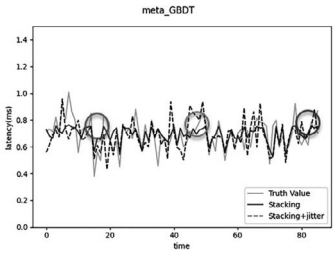

图12 GBDT作为元学习器

Fig.12 GBDT as meta-learner

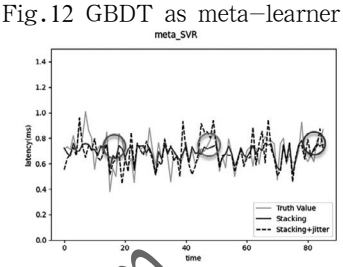

图13 支持向量机回归作为元学习器

Fig.13 SVR as meta-learner

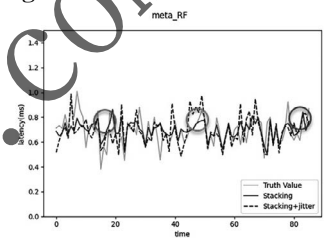

图14 随机森林作为元学习器

Fig.14 RF as meta-learner

从图11—图14中可以看出，每个元学习器在总体上保持一致，但是在细节处理方面(即图中所圈之处)，随机森林的性能更高。由第一层学习器预测到的数据是高度非线性的，随机森林模型在拟合曲线上更加符合真实时延抖动，据此可以说明，利用随机森林作为元学习器符合本文需求。在后续实验中，将选择随机森林作为实验的元学习器。

4.3 数据采样结果影响

为防止模型过拟合，在数据集采样时，借鉴Bagging策略[10]的自助采样法，从所有训练数据中抽取若干子集，每次选取63.2%的数据进行模型训练，未被抽中的36.8%的袋外数据则可以作为验证集对模型进行评估和优化。在验证数据集采样结果对模型训练效果的影响时，采取80%数据的随机采样，既保证了数据的准确性，又保证了由于数据集的不同训练出的模型具有差异性，可以提升本文算法模型的泛化能力。

从图15和图16中可以看出，80%随机取样的数据比100%随机取样的数据作为训练集，在预测效果上更符合真实值，据此可以说明采用自助采样的方式，增加了基学习器之间的差异性，模型泛化能力得到加强。

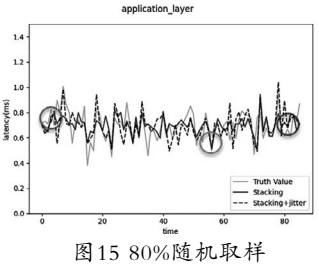

图15 80%随机取样

Fig.15 80% random sampling

(下转第18页)

www.bjgczz.com