文章编号：2095-2163(2023)01-0198-06 中图分类号：TP393.08 文献标志码：A

基于降噪自编码器和弹性网络的入侵检测模型

常会鑫，杨丽敏，陈丽芳
（华北理工大学 理学院，河北 唐山 063210）

摘 要：入侵检测建模中，数据损坏或不完整、训练过程中存在的数据过拟合问题，以及对未知攻击流量缺少判断依据等因素，将影响模型训练效果。针对以上问题，本文提出一种基于降噪自编码器和弹性网络的入侵检测模型。该模型利用降噪自编码器降低输入数据损坏及不完整对模型训练的影响，使用弹性网络解决数据过拟合问题，对未知攻击流量采用损失值作为判断依据。实验结果表明，本文构建的入侵模型与传统机器学习算法及深度学习算法相比具有更高的准确性，检测效果更好。

关键词：降噪自编码器；弹性网络；入侵检测

Intrusion detection model based on noise denoising autoencoder and elastic net

CHANG Huixin, YANG Limin, CHEN Lifang
(School of Science, North China University of Science and Technology, Tangshan Hebei 063210, China)

【Abstract】 In intrusion detection modeling, data corruption or incompleteness will affect the model training effect. There is a problem of data overfitting in the training process. In addition, there is a lack of judgment basis for unknown attack traffic. To solve the above problems, in this paper, an intrusion detection model based on denoising autoencoders and elastic net is proposed. The model uses the denoising autoencoders to reduce the impact of input data damage and incompleteness on model training, uses the elastic net to solve the problem of data overfitting, and uses the loss value as the judgment for unknown attack traffic. In accordance, the experimental results show that the intrusion model constructed in this paper has higher accuracy and better detection effect than traditional machine learning algorithms and deep learning algorithms.

【Key words】 denoising autoencoder; elastic net; intrusion detection systems

0 引 言

近年来，计算机网络发展迅速，对信息化生活产生了重大影响，然而计算机网络的广泛应用使其面临各种严重的威胁，如恶意活动、网络入侵和网络犯罪[1]。入侵检测系统是目前最有前景的网络安全防御机制之一，在网络安全领域引起了大量的关注与研究[2]。尽管入侵检测系统已经发展到一个高度成熟的水平，但由于当今互联网的广泛应用，迅猛增长的网络流量和复杂的网络结构为入侵检测带来新的挑战。如何从海量网络数据中学习最鲁棒的特征表示，提高入侵检测精度是目前的研究热点。

机器学习算法应用，在入侵检测领域使其具有更高的鲁棒性和适应性[3]。许多机器学习算法，如 SVM[4]、logistic 回归[5]、XGBoost[6] 等，已被用于开发入侵检测模型。然而，网络流量数据的高维性和复杂性，使网络入侵检测成为一项具有挑战性的任务。深度学习作为一种机器学习方法，因其对高维大规模数据的挖掘能力而受到广泛关注，成功地解决了文本分类、目标识别、图像分类等研究领域所面临的许多问题，也逐渐应用于网络入侵检测系统中。虽然深度学习算法 RNN[7]、DNN[8] 在入侵检测方面取得了良好的效果，但通过研究发现在其算法中还存在数据不完整对模型训练结果影响过大、训练过程中存在数据过拟合、模型对未知攻击流量缺少判断依据等问题。

为解决上述问题，本文提出了一种基于降噪自编码器（Denoising Autoencoders，DAE）和弹性网络（Elastic Net，EN）的入侵检测模型—DAE-EN。该模型可以高效准确地对网络流量中的攻击流量部分进行识别，达到保证网络安全的目的。为了方便阅读本文以 DAE-EN 代替降噪自编码器-弹性网络模型。

作者简介：常会鑫（1995-），男，硕士研究生，主要研究方向：网络安全；杨丽敏（1997-），女，硕士研究生，主要研究方向：迁移学习、住侵检测；陈丽芳（1973-），女，博士，教授，主要研究方向：数据挖掘。
通讯作者：陈丽芳 Email：hblg_clf@163.com
收稿日期：2022-06-15

1　理论基础

1.1　自编码器概念

自编码器是神经网络算法的一种，用于在无监督的情况下学习有效的数据编码[9]。自编码器为了学习一组数据编码，需要训练网络去除信号中的“噪声”，其主要功能是降低数据特征的维数。自编码器包含编码器和解码器，分别用于将输入映射到隐藏层以及从隐藏层映射到输出。

最简单的自编码器形式是前馈非递归神经网络，其类似于单层感知器，自编码器输入层和输出层需要一个或多个隐藏层进行处理。输入层和输出层必须具有相同数量的节点（神经元）才能保证重构输入，重构的结果是得到一个输入和输出之间的最小化差异，而不是预测给定输入 X 情况下的目标值 Y。

将自编码器中的编码器定义 ϕ，解码器定义为 ψ，则表达式为：

$$\begin{aligned}&\phi: X \to F\\&\psi: F \to X\\&\phi,\psi = \arg\min_{\phi,\psi} \| X - (\psi, \phi)X \|^2\end{aligned} \tag{1}$$

自编码器架构如图 1 所示。通常情况下，自编码只会使用一个隐藏层来连接输入与输出，编码器接收输入 $x \in \boldsymbol{R}^d = X$，并将其映射到 $h \in \boldsymbol{R}^p = F$：

$$h = \sigma(\boldsymbol{W}x + \boldsymbol{b}) \tag{2}$$

式中，h 通常被称为潜在变量；σ 为激活函数，例如 sigmoid 函数或修正线性单元；$\boldsymbol{W}$ 为权重矩阵；$\boldsymbol{b}$ 为偏置向量。权重矩阵和偏置向量的初始化过程是随机的，这两个参数在训练过程中通过反向传播迭代更新。自编码器的解码阶段将 h 映射到与 x 具有相同维度的 x' 为

$$\boldsymbol{x}' = \sigma'(\boldsymbol{W}'h + \boldsymbol{b}') \tag{3}$$

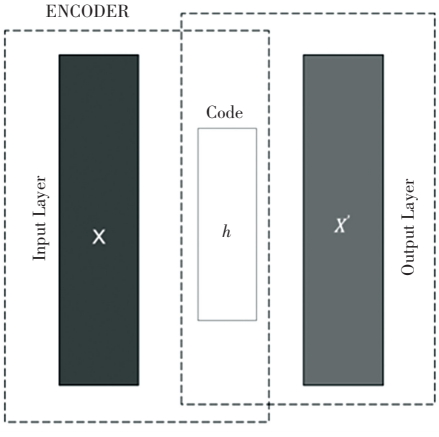

图 1　自编码器架构

Fig. 1　Autoencoder architecture

解码器中的变量 $\sigma'\ \boldsymbol{W}'\ \boldsymbol{b}'$ 与编码器中的变量 $\sigma\boldsymbol{W}\boldsymbol{b}$ 没有强关联性。自编码器经过训练最小化重建误差（例如：平方误差），通常称为“损失”：

$$L(\boldsymbol{x},\boldsymbol{x}') = \| \boldsymbol{x} - \boldsymbol{x}' \|^2 = \| \boldsymbol{x} - \sigma'(\boldsymbol{W}'(\sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})) + \boldsymbol{b}') \|^2 \tag{4}$$

X 通常是训练集的平均值。自编码器训练与其它前馈神经网络一样，均是通过误差的反向传播进行。

特征空间 F 的维度应低于输入特征 X 的维度，特征向量 $\boldsymbol{\phi}(X)$ 可以看做是输入 x 压缩而来的。在使用欠完备自编码器情况下，如果隐藏层节点数量大于等于输入层的节点数量，自编码器会一直学习同一个函数，导致整个训练过程是无效的。然而通过实验发现，欠完备的自编码仍然可能学习到有用的特性[10]，模型代码维度和模型容量可以根据所需建模的数据分布情况来设定，这种方法称为正则化自编码器。

1.2　降噪自编码器

降噪自编码器（DAE）是在自编码器的基础上增强了健壮性[11]，DAE 使用部分“损坏”的输入并经过训练以恢复原始未失真的输入。在实践中，自编码器去噪的目标是清除“损坏”的输入，为实现这个目的通常采用两种方法[12]：一是使用更高级别的输入特征，提高相对稳定性和健壮性；二是为了更好的对特征进行去噪，模型需要提取特征来分析输入中的有效结构。DAE 主要是为了从噪声输入数据中学习更一般化的特征，其训练过程如下：

Step 1　通过随机映射，破坏初始输入 x 使其变为 $\tilde{x}$，公式为：$\tilde{\boldsymbol{x}} \sim \boldsymbol{q}_D(\tilde{\boldsymbol{x}} \mid \boldsymbol{x})$。

Step 2　将 $\tilde{x}$ 映射到隐藏层，其过程与常规自编码器一致，$h = \boldsymbol{f}_\theta(\tilde{\boldsymbol{x}}) = \boldsymbol{s}(\boldsymbol{W}\tilde{\boldsymbol{x}} + \boldsymbol{b})$。

Step 3　在隐藏层中对该模型进行重构 $z = g_{\theta'}(h)$。

模型中的参数 θ 和 θ' 是为了最小化训练集的平均重构误差，特别是最小化数据 z 和原始未输入数据 $\boldsymbol{x}$ 之间的差异。基于 $\boldsymbol{q}_D(\tilde{\boldsymbol{x}} \mid \boldsymbol{x})$，每个随机示例 X 输入到模型中，都会随机产生一个新的“损坏”版本。

1.3　弹性网络

在统计学中，特别是在线性或逻辑回归模型的拟合中，弹性网络线性组合了 Lasso 和 Ridge 方法的 L_1 和 L_2 惩罚[13]。弹性网络可以向 Lasso 学习输入少量参数非零的稀疏模型，同时能保持 Ridge 的正则性质。该方法基于公式（5）：

$$\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j| \boldsymbol{x}' = \sigma'(\boldsymbol{W}'h + \boldsymbol{b}') \quad (5)$$

在高维数据、样本数量不多的情况下,Lasso 在饱和之前最多选择 n(样本个数)个变量[14]。此外,若样本是一组高度相关的变量时,Lasso 会优先选择变量中的一个变量而忽略其它变量。为了克服这些限制,弹性网络增加了一个惩罚 $\|\beta\|^2$,单独使用时就是一个 Ridge 回归[15]。弹性网络的定义为

$$\hat{\beta} \equiv \text{argmin}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1) \quad (6)$$

二次惩罚项使损失函数具有很强的凸函数性质,因此具有唯一的最小值[16]。

1.4 DEA-EN 模型构建与实现流程

DEA-EN 模型的构建主要是由两部分组成:一是由输入层、隐藏层、输出层组成的降噪自编码器(DEA);二是在 DEA 的隐藏层中增加的弹性网络。

如图 2 所示,DAE-EN 模型的实现步骤如下。

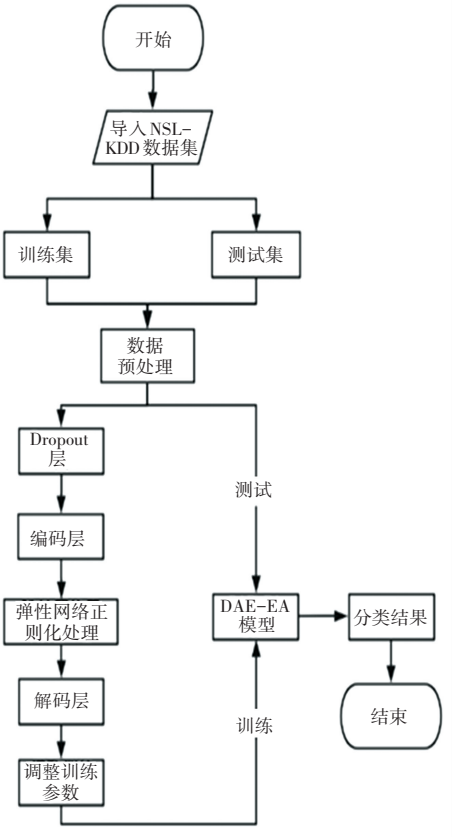

图 2 DAE-EN 模型实现流程

Fig. 2 DAE-EN model implementation process

Step 1 导入 NSL-KDD 数据集,按照 8:2 的比例将该数据集划分训练集和测试集。为方便模型使用,训练集和测试集均需要对数据进行预处理。

Step 2 将训练集数据输入 Dropout 层,其可以随机的临时屏蔽掉一半的隐藏神经元,达到防止模型过拟合,提升模型泛化能力。

Step 3 经过一个具有 8 个神经元的编码层进入降噪自编码器的隐藏层。在隐藏层采用弹性网络,避免数据的过拟合,最后再从解码层输出,最终构建好整个模型。

Step 4 调整模型参数。

Step 5 通过训练集数据对模型进行训练,使用测试集评估模型效果,得到最终分类结果。

2 仿真实验

2.1 数据集来源

NSL-KDD 数据集基于 KDD99 进行改进,解决了 KDD99 数据集数据的重复性和冗余性问题,避免了训练模型优先使用出现频率较高数据的情况,该数据集被广泛应用于入侵检测系统中。NSL-KDD 数据集中共有 148 517 条数据,训练集和测试集占比为 84.8% 和 15.2%,分别存放在 NSL_Trian.csv 和 NSL_Test.csv 文件中。整个数据集包含 41 个特征值和一个标志位(见表 1),同时 NSL-KDD 数据集包含 39 种攻击类型,按照其性质划分为 4 种一般的攻击类型:DoS、R2L、U2R 和 Probe。具体分类情况见表 2。

表 1 NSL-KDD 数据集的部分数据

Tab. 1 Partial data of NSL-KDD dataset

NO	duration	protocol_type	service	...	outcome	difficulty
0	0	tcp	private	...	neptune	21
1	0	tcp	private	...	neptune	21
2	2	tcp	ftp_data	...	normal	21
...	...	...	...	...	...	...
22540	0	tcp	http	...	back	15
22541	0	udp	domain_u	...	normal	21
22542	0	tcp	sunrpc	...	mscan	14

表 2 NSL-KDD 数据集中的攻击分类

Tab. 2 Attack classification in NSL-KDD dataset

攻击类型	攻击名字
DoS	Snmpgetattack back land neptune smurf teardrop pod apache2 udpstorm processtable mailbomb
R2L	snmpguess worm httptunnel named xlock xsnoop sendmail ftp_write guess_passwd imap multihop phf spy warezclient warezmaster
U2R	sqlattack buffer_overflow loadmodule perl rootkit xterm ps
Probe	ipsweep nmap portsweep satan saint mscan

2.2　数据集预处理

数据集预处理是模型训练前的必要步骤，其中包括两部分处理内容：针对连续数据进行归一化处理；针对字符串数据进行一位有效编码处理。

采用最小-最大归一化方法，将连续值缩放到数值范围[0,1]，如公式(7)：

$$\tilde{\boldsymbol{x}}_{fj} = \frac{\boldsymbol{x}_f - \min(\boldsymbol{x}_f)}{\max(\boldsymbol{x}_f) - \min(\boldsymbol{x}_f)} \tag{7}$$

其中，$\max(\boldsymbol{x}_f)$ 和 $\min(\boldsymbol{x}_f)$ 分别代表特征值 $\boldsymbol{x}_{fj}$ 的最大值和最小值。归一化的结果 $\tilde{\boldsymbol{x}}_{fj}$ 取值范围为[0,1]。

一位有效编码将协议类型、服务和标志 3 个特征转换为数值，每个类别属性都由二进制值表示。例如：protocol_type 字段有 tcp、udp 和 icmp 3 个值，一位有效编码将其分别转换为二进制向量[1,0,0]、[0,1,0]、[0,0,1]，同时把服务和标志两个字段处理成向量。经过预处理后，数据集的特征值由原来的 41 个变成了 122 个。其中，连续特征 38 个，协议类型、服务和标志 3 个特征相关的向量特征为 84 个。

2.3　训练方法及参数调整

为了避免训练数据中每种攻击类型样本不平衡以及无法通过训练分析未知攻击类型的问题，本文提出利用自编码器和弹性网络来检测异常的方法。该方法使用一个带有 dropout 的降噪自编码器。由于输入样本的特征数是 122，所以输入层由 122 个神经元组成，紧接着是一个 dropout 层和一个由 8 个神经元组成的隐藏层，最后是一个有 122 个神经单元的输出层，隐藏层和输出层的激活函数为 relu 函数。

降噪自编码器通过训练，将输入进行重构。模型只针对训练集中标志位为正常的数据训练，进一步得到输入和输出之间的均方误差，最终目的是通过训练使均方误差最小化。自编码器强制在训练阶段结合弹性网络正则化处理，防止训练时出现简单地将输入复制到输出的情况。为防止过拟合问题，需要对 Lasso、Ridge 的 L_1 和 L_2 惩罚参数进行调整，均设为 0.001时效果最好。此外，这种自编码器被训练成从自身的一个“损坏”版本来重构输入，迫使自编码器学习更多的数据属性。该模型使用大小为 100 的 Adam 优化器训练 10 个 epoch，使用训练集 90%的数据对模型进行训练，预留 10%的数据验证模型。

2.4　结果分析

本模型的训练方法只使用正常样本数据，将正常数据与攻击数据分别输入到训练好的模型中。具有攻击的数据所得到的重构误差相对偏高，因此可以通过设置重构误差的阈值检测攻击，假设一个数据样本的重构误差高于预设的阈值，该样本则被归为攻击，否则归为正常流量。

对于阈值的选择，采用训练数据上的模型损失和验证数据上的模型损失，本实验中使用训练数据生成的模型损失作为一个阈值。

本次实验的目的是为了进行异常检测，检测网络数据中的攻击数据，本模型只涉及二分类。为了更好的评估模型的性能，根据输入特征与预测特征的特性，使用了一个损失函数式(8)计算每个输入的重构损失，通过对比重构误差和预设阈值的大小来进行分类。

$$MSE(\boldsymbol{y},\boldsymbol{y}') = \frac{\sum_{j=1}^{n} (\boldsymbol{y}_i - \boldsymbol{y}_i')^2}{n} \tag{8}$$

为了评估该模型，准确率(*Accuracy*)、精确率(*Precision*)、召回率(*Recall*)、*F* 分数(*F - score*)等计算性能指标。其计算公式分别为：

$$Accuracy = \frac{\sum_{j=1}^{M_{Test}} TP_j}{N_{Test}} \tag{9}$$

$$Precision = \frac{TP_j}{TP_j + FP_j} \quad 1 \leqslant j \leqslant M_{Test} \tag{10}$$

$$Recall = \frac{TP_j}{TP_j + FN_j} \quad 1 \leqslant j \leqslant M_{Test} \tag{11}$$

$$F - score = \frac{2Precision * Recall}{Precision + Recall} \tag{12}$$

通过实验得出准确率为 90.3 ，召回率为 95.3 ，精确率为 88.5，F_1 分数为 91.8。为了使数据更加直观，绘制了正常流量和攻击流量的混淆矩阵[14]，如图 3 所示。其中，横轴是预测值，纵轴是实际值。

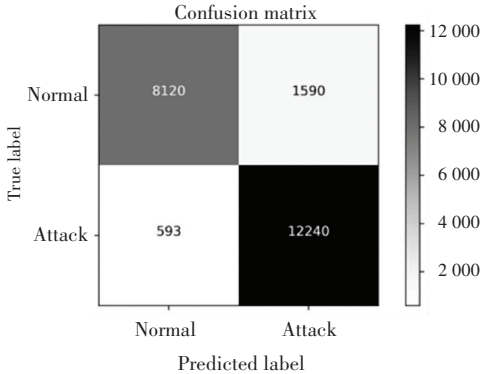

图 3　正常流量与攻击流量的混淆矩阵

Fig. 3　Confusion matrix of normal traffic and attack traffic

针对模型的测试结果，计算出 4 种攻击的检测率见表 3，以及如图 4 所示的小提琴图。由图中可

以清楚的看出测试集中所有数据样本重构损失值的分布。其中,攻击的损失值大多高于阈值,正常流量的值则大部分小于阈值。

表 3 4 种攻击类型检测率

Tab. 3 Detection rates of four attack types

攻击类型	检测率
DoS	0.931
R2L	0.977
U2R	0.955
Probe	0.99

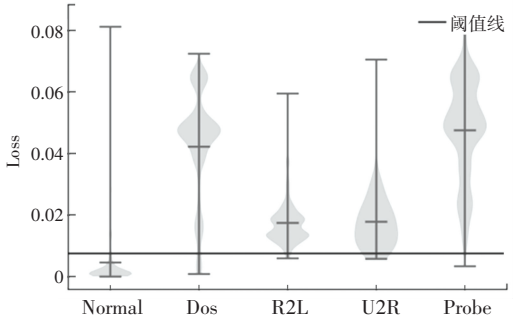

图 4 测试集的损失分布

Fig. 4 Loss distribution of the test set

2.5 算法性能对比分析

为了进一步验证本模型针对入侵检测的性能,将其与决策树(DT)、K 邻近算法(KNN)、梯度提升决策树(GBDT)、随机森林(RF)以及变分自编码器(VAE)算法[15-16]的准确率进行对比,其结果见表 4。

表 4 与传统机器学习算法对比

Tab. 4 Comparison with traditional machine learning algorithms

算法	准确率
DT	81.42
KNN	76.25
GBDT	76.25
RF	70.10
VAE	84.28
DAE-EN	90.31

由表 4 中可以看出,决策树、K 邻近算法、梯度提升决策树、随机森林 4 种传统机器学习算法中,准确率最高的为决策树算法,其值为 81.42%;变分自编码器深度学习模型的准确率为 84.28%;而 DAE-EN 模型对于区分正常流量与攻击流量的能力都优于其它算法,准确率为 90.31%。

3 结束语

入侵检测系统面对大量的网络流量,其中大部分流量是正常流量,只有小部分流量是具有攻击的流量,所以本文设计 DAE-EN 模型通过训练正常流量计算损失值,根据重构误差的大小,通过设置合理的阈值来区分正常数据和攻击数据。DAE-EN 模型中降噪自编码器将输入数据经编码器和解码器映射后,生成重构数据,在训练过程中增加了弹性网络正则方法避免训练过程的过拟合问题。通过实验结果对比表明,本文提出的 DAE-EN 模型检测准确性达到 90.31%,高于决策树等传统机器学习算法以及变分自编码器模型,对网络入侵检测具有更优的检测效果。目前,DAE-EN 模型只能区分正常流量和攻击流量,计划在未来工作中将模型扩展到多个类别,并且可以完成对特定攻击类型的识别。

参考文献

[1] 刘剑, 苏璞睿, 杨珉, 等. 软件与网络安全研究综述[J]. 软件学报, 2018, 29(1): 42-68.

[2] BINBUSAYYIS A, VAIYAPURI T. Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection[J]. Heliyon, 2020, 6(7): e04262.

[3] ALDWEESH A, DERHAB A, EMAM A Z. Deep learning approaches for anomaly - based intrusion detection systems: A survey, taxonomy, and open issues [J]. Knowledge - Based Systems, 2020, 189: 105124.

[4] LIU W, CI L L, LIU L P. A new method of fuzzy support vector machine algorithm for intrusion detection[J]. Applied Sciences, 2020, 10(3): 1065.

[5] MAALOUF M, HOMOUZ D, TRAFALIS T B. Logistic regression in large rare events and imbalanced data: A performance comparison of prior correction and weighting methods [J]. Computational Intelligence, 2018, 34(1): 161-174.

[6] BHATTACHARYA S, MADDIKUNTA P K R, KALURI R, et al. A novel PCA - firefly based XGBoost classification model for intrusion detection in networks using GPU[J]. Electronics, 2020, 9(2): 219.

[7] ALMIANI M, ABUGHAZLEH A, AL-RAHAYFEH A, et al. Deep recurrent neural network for IoT intrusion detection system [J]. Simulation Modelling Practice and Theory, 2020, 101: 102031.

[8] XU C, SHEN J, DU X. A method of few-shot network intrusion detection based on meta - learning framework [J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 3540-3552.

[9] KRAMER M A. Nonlinear principal component analysis using autoassociative neural networks [J]. AIChE journal, 1991, 37(2): 233-243.

[10] BENGIO Y. Learning deep architectures for AI[J]. Foundations and trends® in Machine Learning, 2009, 2(1): 1-127.

[11] VINCENT P, LAROCHELLE H, BENGIO Y, et al. Extracting and composing robust features with denoising autoencoders[C]// Proceedings of the 25th international conference on Machine learning. 2008: 1096-1103.

(下转第 207 页)