北京大学

二学位毕业论文

题目：基于聊天信息的持续身份认证机制的设计与实现

姓　　名：丁笠峰

学　　号：

院　　系：软件与微电子学院

专　　业：软件工程二学位

导师姓名：

二〇二五 年 六 月

版权声明

摘要

为解决用户在线会话期间缺乏持续的身份验证手段保障用户账号安全的问题，本文对比了不同的持续认证方案的优点和不足，设计并实现了一种基于聊天信息的持续身份认证机制。该机制通过分析用户历史聊天记录与新输入的聊天文本间在用词习惯、造句方式和内容偏好等特征上的相似性，认证用户的身份。用户认证失败时，机制会登出用户，要求用户再次输入用户名和密码验证身份。

本文通过实验比较不同组合模型的性能后，最终选择基于 BERT(Bidirectional Encoder Representations from Transformers)的组合模型设计和实现持续身份认证机制，基于公开数据集的测试结果显示，模型的 F1 值为 0.77，ROC 曲线的平均 AUC 值为 0.8，机制具备基本的用户身份认证能力。在 Fast API 框架上部署持续身份认证机制开展性能测试，结果显示在每秒 100 个并发请求、持续 10 分钟的测试场景下，持续认证机制的吞吐量为 6.65rps，平均响应时间为 13.32 秒，最大响应时间不超过 3 分钟。

相比其他持续身份认证机制，本文设计的持续身份认证机制认证效果稳定、用于认证判断的数据容易获取，能够便捷地部署在具有聊天服务的系统中，具有一定的实际应用价值。

关键词：持续认证机制，聊天信息，BERT 模型，聊天服务

Design and Implementation of a Continuous Authentication Mechanism Based on Chat Information

Lifeng Ding (Software Engineering)
Directed by Huiping Sun

ABSTRACT

To address the issue of insufficient continuous authentication methods during online sessions for ensuring user account security, this paper compares the advantages and limitations of various continuous authentication schemes and proposes a continuous authentication mechanism based on chat information. The mechanism authenticates user's identities by analyzing the similarity between historical chat records and newly input chat texts in terms of word usage habits, sentence construction patterns, and content preferences. when authentication fails, it logs the user out and requires re-authentication.

After experimentally comparing the performance of different combined models, we ultimately selected a BERT (Bidirectional Encoder Representations from Transformers)-based composite model to design and implement the continuous authentication mechanism. Test results on a public dataset show that the model achieves an F1-score of 0.77 and an average AUC value of 0.8 for the ROC curve, demonstrating its fundamental capability for user identity authentication. Deployed on the Fast API framework, performance tests under a scenario of 100 concurrent requests per second over a 10-minute duration reveal a throughput of 6.65 requests per second, an average response time of 13.32 seconds, and a maximum response time under 3 minutes.

Compared to other continuous authentication mechanisms, the proposed solution exhibits stable authentication performance, leverages easily accessible data for authentication decisions, and can be seamlessly integrated into systems with chat services. This highlights its practical value for real-world applications.

KEY WORDS: Continuous authentication mechanism, Chat information, BERT, Chat services

目录

摘要 I
ABSTRACT II
目录 III

第一章 绪论 1
1.1 研究背景 1
1.2 研究现状 2
1.2.1 持续身份认证机制 2
1.2.2 基于传统机器学习算法的文本作者识别技术 2
1.2.3 基于深度学习算法的文本作者识别技术 3
1.3 本文主要工作 4
1.4 本文结构 5

第二章 需求分析 6
2.1 功能性需求概述 6
2.2 用户身份认证需求 7
2.2.1 用户身份认证模型 7
2.2.2 判断阈值 7
2.2.3 认证结果的处理方案 8
2.3 用户文本特征更新需求 8
2.4 非功能性需求 8
2.5 本章小结 9

第三章 持续身份认证机制的概要设计 10
3.1 持续身份认证机制架构设计和功能模块 10
3.1.1 持续身份认证机制架构 10
3.1.2 持续身份认证机制的功能模块 11
3.2 用户身份认证模型设计 12
3.2.1 不同模型组合的认证效果比较 12
3.2.2 基于 BERT 的短文本特征提取 13
3.2.3 双向 GRU 网络设计 13
3.2.4 注意力机制 14
3.2.5 softmax 函数 15

3.3 判断阈值和认证结果处理方案设计 16
3.3.1 判断阈值设计 16
3.3.2 认证结果处理方案设计 16
3.4 用户文本特征更新机制设计 17
3.4.1 存储用户聊天信息的数据库的设计 17
3.4.2 文本特征更新机制 17
3.5 本章小结 18

第四章 持续身份认证机制的详细设计与实现 19
4.1 开发环境 19
4.2 持续身份认证机制整体的设计与实现 19
4.3 用户身份认证模型的设计与实现 20
4.3.1 用户身份认证模型的实现 21
4.3.2 构建数据集微调模型 22
4.4 判断阈值和认证结果处理方案的设计与实现 24
4.5 用户文本特征更新机制 25
4.5.1 存储用户聊天信息的数据库的设计与实现 26
4.5.2 文本特征更新机制的设计与实现 27
4.6 本章小结 28

第五章 持续身份认证机制测试 29
5.1 实验和测试设计 29
5.1.1 功能性测试设计 29
5.1.2 非功能性测试设计 29
5.2 测试环境和测试数据 30
5.2.1 测试环境与测试工具 30
5.2.2 公开数据集的基本情况 31
5.2.3 认证模型及判断阈值测试的测试数据 33
5.3 功能性测试结果 34
5.3.1 认证模型及判断阈值的测试结果 34
5.3.2 认证结果处理功能的测试结果 35
5.3.3 文本特征更新功能的测试结果 36
5.4 非功能性测试结果 37
5.4.1 兼容性测试结果 37
5.4.2 吞吐量测试结果 37
5.4.3 响应时间测试结果 38

5.5 本章小结 38
第六章 结论与展望 39
6.1 结论 39
6.2 展望 39
参考文献 40
致谢 42

第一章　绪论

基于聊天信息的持续身份认证机制，是一种在用户使用聊天服务的过程中，基于用户输入的聊天文本中隐含的用词习惯、造句方式和内容偏好等特征，持续地验证用户身份的机制。

1.1 研究背景

近年来，在线社交平台的用户量快速增长，根据腾讯控股的财报，截止 2024 年 12 月 31 日，微信及 WeChat 平台每月活跃账户数为 13.85 亿，QQ 移动终端每月活跃账户数为 5.24 亿。[1]

同时，移动支付功能的使用率也大幅增加，根据中国互联网络信息中心（CNNIC）第 54 次《中国互联网络发展状况统计报告》，截止 2024 年 6 月，我国网络支付用户规模达 9.6885 亿人。[2]根据咨询平台公开数据，2024 年我国移动支付用户常用的手机支付平台中，微信、支付宝和云闪付位列前三。其中，微信和支付宝等主流平台，将社交、购物、小程序和移动支付等功能高度集成，用户可在聊天窗口中便捷地购买商品和发起转账。

但是在线社交平台和移动支付功能的蓬勃发展在方便用户操作的同时，也增加了用户账户的安全风险。攻击者能够通过冒充用户家人或好友的方式，诱导用户转账和获取用户隐私信息。

传统的身份认证方式，如用户名密码、双因素认证等，虽然能够在用户首次登陆时有效地验证用户身份，但在用户在线聊天的过程中却缺乏持续的身份验证手段保障用户的账号安全。

为解决这一问题，研究者引入了持续身份认证机制，通过在用户聊天的过程中不断地认证用户身份，弥补传统的身份认证方式的不足。同时，以往的关于持续身份认证机制的研究主要使用生物特征进行身份认证，基于聊天信息进行持续身份认证的研究相对较少。[3]

使用生物特征进行持续身份认证，面临着很多的挑战，例如生物特征模板一旦泄露无法更改，可能导致用户身份被永久盗用；其次，持续地监控摄像头或麦克风会降低用户对持续认证机制的接受度，提高用户对于自身隐私泄露的担忧。[4]

相较而言，基于聊天信息的持续身份认证机制，具有不采集用户的生物特征数据、不使用摄像头和麦克风等专用硬件、交互自然等优势，能够根据用户输入的聊天文本持续验证用户身份，确保用户在线会话期间的身份一致性。在增加系统安全性的同时，

减少用户使用过程中受到的打扰，为在线社交平台用户提供安全流畅的使用环境。

1.2 研究现状

基于聊天信息的持续身份认证机制可以视作持续认证机制和文本作者识别两种技术的结合，因此本节将从持续认证机制的研究现状和文本作者识别技术的研究现状两个方面来介绍。

1.2.1 持续身份认证机制

Baig AF 等人[4]将持续身份认证机制按照认证方式分类为基于生物特征的持续身份认证机制、基于行为特征的持续身份认证机制和基于情境感知的持续身份认证机制三类，下文将照此分类介绍持续身份认证机制的研究现状。

基于生物特征的持续认证机制主要通过指纹、人脸、声音和虹膜等特征进行用户认证，其中 Crouse 等人[5]在 2015 年使用 SVM（Support Vector Machine，支持向量机）分类算法在移动设备上实验了基于人脸识别的持续身份认证机制；Feng 等人[6]使用 SVM 分类算法实现了一种基于声音识别的持续身份认证机制。但是，此类基于生物特征的持续认证机制，涉及用户敏感信息的获取和监测，在实际应用中面临着很多挑战。

基于情境感知的持续身份认证机制利用 IP 地址、GPS 定位和电池使用情况等在时间顺序上前后关联的数据进行用户认证，具体而言，Gomi 等人[7]基于用户浏览历史中的 IP 地址、URL 和访问时间，使用 LR（Linear Regression，线性回归）模型分类，实现了一种基于用户浏览记录的持续身份认证机制。不过这种认证机制依赖的情境数据易被伪造和篡改，在用户行为或用户活动环境改变时，认证能力显著下降。

基于行为特征的持续身份认证机制通过用户的步态特征、键盘输入时的击键动态、用户输入文本的特征和用户的眼动特征等行为进行用户认证。具体而言，Derawi 等人[8]通过手机中的运动传感器采集的用户步态特征，进行用户认证；Brocardo 等人[9]通过验证用户输入文本的特征，使用 SVM 分类算法实现用户认证；Gascon 等人[10]使用 SVM 分类算法和用户编写短信时的击键动态，实现了一种持续认证机制。

其中，基于用户输入文本的特征进行持续身份认证的方案，具有认证效果稳定（用户的文本特征相对固定）和数据容易获取等特点；因此，本文计划设计并实现一种基于聊天信息的持续身份认证机制，保障用户使用在线聊天服务期间身份的一致性。

1.2.2 基于传统机器学习算法的文本作者识别技术

通过文本信息分析作者身份的研究起源于语言学领域关于文体的归纳分析[11]，根据张洋[12]等人的作者识别研究综述，基于聊天信息的用户身份认证的流程可以分为两

步，首先是从聊天信息的文本数据中提取特征，然后使用分类器对文本数据进行分类，判断文本是否由同一用户编写。

首先介绍基于传统机器学习算法的文本作者识别技术，针对网络上文本信息的作者身份识别，Zheng 等[13]研究者通过选取词法特征、句法特征等构成特征集，采用 SVM 等分类算法对实验语料进行作者身份识别取得了明显的区分效果；吕英杰等人[14]使用词汇特征、句法特征、结构特征和内容特征，针对中文文本信息的特点，使用决策树和支持向量机等分类算法对 BBS 论坛文本和博客文本进行作者识别，在少量作者的情况下取得了良好的区分效果。

但是，上述方法依赖于文本的字词频率、功能词数量、标点数量等统计特征，不适合使用在文本长度短、文本格式随意的聊天信息文本的识别中。在此基础上，部分研究者尝试通过复杂的特征提取，实现短文本的作者身份识别，如祁瑞华等人[15]通过结构、句法和词汇建立多层面博客文本文体特征模型，Yang 等人[16]提出一种主题漂移模型，通过分析文本的写作风格和主题实现作者识别。

1.2.3 基于深度学习算法的文本作者识别技术

基于深度学习算法的文本作者识别技术使用神经网络搭建分类器，例如徐晓霖等人[17]提出 CABLSTM 中文微博作者身份识别模型，通过最大化提取短文本特征、融合 attention 机制于 CNN 中、去除池化层和使用双向 LSTM 获取上下文信息，将身份识别结果通过 softmax 层进行输出；实验结果显示，模型在中文微博作者身份识别任务中的准确率、召回率和 F 值相比传统机器学习算法、TextCNN 算法和 LSTM 算法都取得了较大的提升。模型结构如图 1.1 所示。

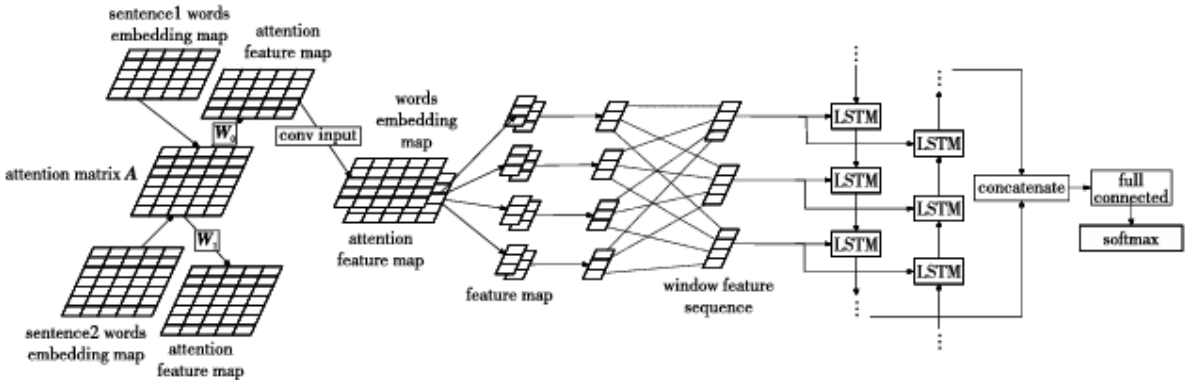


图 1.1 短文本识别 CABLSTM 模型结构[17]

Zhang 等人[18]提出了一种语法编码方案，为句子中的每个单词构建一个嵌入向量，编码每个单词在语法树中的路径，并将获得的嵌入向量输入 CNN 模型中实现作者识别，取得较好的成效。

冯勇等人[19]提出了融合 TF-IDF 和 LDA 的中文 FastText 短文本分类方法，该方法

在 FastText 文本分类模型的输入阶段对 n 元语法模型处理后的词典进行 TF-IDF 筛选，并使用 LDA 模型进行语料库主题分析，确保计算输入词序列向量均值时偏向高区分度的词条，使 FastText 文本分类模型更适用于中文短文本分类环境。

张翼翔[20]等人首先使用 BERT 模型提取文本特征生成词向量，然后使用 BiGRU 模型理解文本信息，同时引入注意力机制优化 BiGRU 的输出，最后使用 A-softmax 输出文本属于特定作者的概率。模型结构如图 1.2 所示。

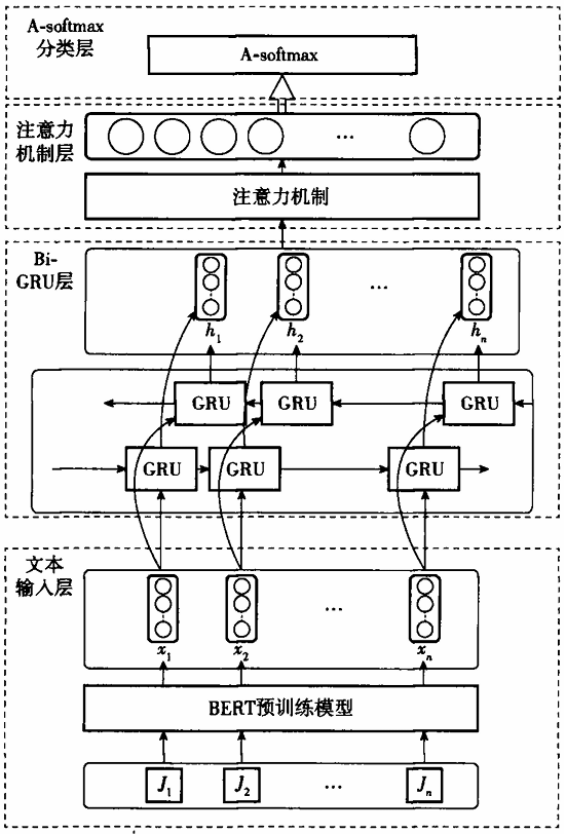


图 1.2 BERT-BiGRU-ATT 模型结构[20]

其中，使用 BERT 模型提取文本特征生成词向量的方案，具有特征提取简单和识别效果优秀的特点。因此，本文将基于该模型结构进行微调和实验，实现一种基于聊天信息的持续身份认证机制。

1.3 本文主要工作

本文设计实现的基于聊天信息的持续身份认证机制的主要工作如下：

（1） 用户身份认证功能

开展实验对比不同模型组合基于聊天信息认证用户身份的能力，根据实验结果，设计一个用户身份认证模型，使用新加坡国立大学收集的中文短信数据集（NUS SMS

Corpus）[21]进行训练和测试。

根据测试结果绘制 ROC（Receiver Operating Characteristic Curve，受试者工作特征曲线）曲线，选择 ROC 曲线上 TPR（True Positive Rate，真正类率）与 FPR（False Positive Rate，假正类率）的差最大的点的索引值作为判断聊天信息是否属于用户的判断阈值。

（2） 更新用户文本特征

用户的文本特征不是一成不变的，其聊天主题、使用的表情包（Emoji 表情）、语气词等会随着热点信息的变化和用户的心情发生改变。

当用户认证失败，触发验证机制并正确的输入用户名及密码后，将触发验证的聊天信息文本添加到用户的历史聊天记录中，实现用户文本特征的更新。

（3） 基于聊天信息的持续身份认证机制的实现

组合上述模块，实现基于聊天信息的持续身份认证机制。使用该机制的系统在用户发出聊天信息的同时，会同步地提交聊天信息至用户身份认证模型，根据模型的输出结果，持续认证机制决定是否要对当前用户发起用户名-密码验证。

1.4 本文结构

本文总共分为六个章节，除绪论外，本文的核心内容按照需求分析、概要设计、详细设计与实现、测试、结论与展望的顺序书写。

在第二章需求分析中，本文分析了持续身份认证机制的功能性需求和非功能性需求。在功能性需求部分，本文描述了机制的核心功能模块应该满足的要求。在非功能性需求部分，本文讨论了机制在实际应用中应满足的兼容性、吞吐量和响应时间要求。

在第三章概要设计中，本文从持续身份认证机制的架构设计和功能模块展开，介绍了用户身份认证功能和文本特征更新功能两个主要功能模块的技术选型。

在第四章详细设计与实现中，本文首先简单介绍了持续身份认证机制整体的设计与实现，方便读者理解机制的主要流程。然后，本文从用户身份认证功能和文本特征更新功能两个方面，补充说明了机制的实现细节。

在第五章测试中，本文按照需求分析的要求，使用公开数据集开展实验，从功能性测试和非功能性测试两个方面，全面地测试本文设计的持续身份认证机制。

在第六章结论与展望中，作者总结了本文的主要工作和测试结果，分析了本文工作的局限性和未来的研究方向。

第二章　需求分析

本章节首先对用户身份认证和根据聊天信息更新用户的文本特征两个核心功能模块，进行了需求分析。然后，本章在非功能需求部分，讨论了基于聊天信息的持续身份认证机制在实际应用中应满足的兼容性、吞吐量和响应时间要求。

2.1 功能性需求概述

在线聊天系统的主要角色包括用户和管理员，其中用户被动地使用基于聊天信息的持续认证机制验证自己的身份。

基于聊天信息持续认证用户身份的用例包含用户身份认证和更新用户文本特征两个用例，下文也将从用户身份认证需求和用户文本特征更新需求两个方面介绍本文设计的持续认证机制的功能性需求。

用户身份认证用例包含编码输入文本、计算置信度和判断认证结果三个用例。当用户认证失败时，判断认证结果用例将扩展弹出二次验证界面用例，要求用户再次输入用户名和密码验证用户身份。

当用户在短时间内通过二次验证时，更新用户文本特征用例扩展修改文本标签用例，修改数据库中导致用户认证失败的聊天文本的标签，更新用户的历史聊天记录。当用户触发二次验证，但经过较长的时间才再次登陆系统时，更新用户文本特征用例扩展删除文本用例，删除数据库中导致用户认证失败的聊天文本，更新用户的历史聊天记录。

本文设计的基于聊天信息的持续身份认证机制的用例图如图 2.1 所示。

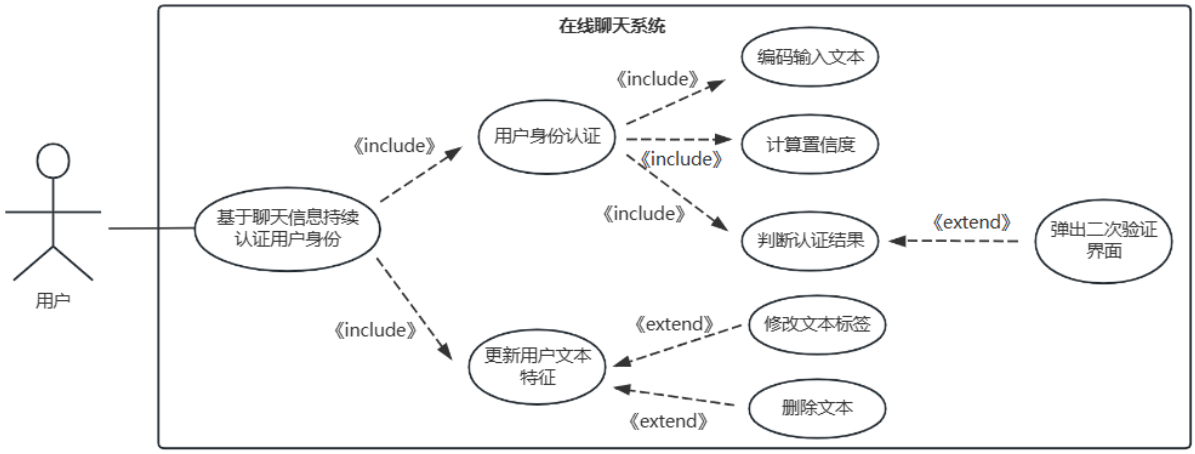


图 2.1 基于聊天信息的持续身份认证机制的用例图

2.2 用户身份认证需求

用户身份认证功能是基于聊天信息的持续身份认证机制的核心功能。在接收聊天服务发送的用户识别码、用户新发送的聊天信息和用户的历史聊天记录后，用户身份认证功能通过用户身份认证模型计算聊天信息属于当前用户的概率，并将计算结果与认证机制设定的阈值对比。

当模型计算的概率低于阈值时，用户身份认证失败，持续认证机制通知系统登出用户并跳转至登陆验证界面；当模型计算的概率高于阈值时，用户身份认证成功，用户能够继续正常使用聊天服务。

为了实现用户身份认证功能，首先需要制作一个用户身份认证模型，计算用户新发送的聊天信息和历史聊天记录由相同作者产生的概率；然后根据用户身份认证模型的测试结果，选择合适的判断阈值，判断用户是否认证成功；最后，需要在聊天服务中实现一个登出指定用户的功能。

综上所述，用户身份认证功能的需求将分成三个部分进行分析，分别说明用户身份认证模型的需求、判断阈值设定的需求和认证结果处理的需求。

2.2.1 用户身份认证模型

首先分析用户身份认证模型，身份认证模型是一个判别式模型，由不同的模型组合形成。在将聊天服务转发的文本数据编码为张量后，模型组首先使用 BERT 模型提取短文本的特征为 768 维向量，然后使用 BiGRU 模型继续提取数据中的信息，并引入注意力机制优化提取到的特征张量。最后，使用 softmax 函数处理身份认证模型全连接层的输出结果，得到用户新发送的聊天信息由用户本人编写的概率。

为保证基于聊天信息的用户认证模型能够较好的区分用户本身和冒用者，应通过实验观察模型处理用户认证任务的查全率。根据张翼翔[20]等人使用 BERT-BiGRU-ATT 模型结构对微博用户数据进行的用户身份识别的实验结果，在用户数量多、用户发布的文本数量平均值小的前提下，使用同类模型结构进行用户身份认证的查全率应在 0.7 以上。

为保证正常用户使用过程中不受验证机制的频繁打扰，应通过实验观察模型处理用户认证任务的查准率。根据李孟林[22]等人使用 CNN-BiLSTM-Attention 模型结构对相同数据集（中文短信数据集 NUS SMS Corpus）进行测试的结果，在模型训练轮数有限的前提下，基于聊天信息的用户认证模型的查准率应在 0.7 以上。

2.2.2 判断阈值

具体而言，经过 softmax 函数处理的用户身份认证模型的全连接层输出并不等于新输入的聊天文本属于当前用户的真实概率，所以不应直接使用 0.5 作为判断聊天文本是

否属于当前用户的阈值。

为保证选定的判断阈值能够有效地区分用户本身和冒用者，应考虑使用用户身份认证模型测试结果中 TPR-FPR 最大的点对应的分类阈值作为判断阈值。在 ROC 曲线中，每个点对应不同分类阈值下的 TPR 和 FPR，其中 TPR-FPR 最大的点等价于 TPR+TNR（True Negative Rate，真负类率）最大的点，所以该点对应的分类阈值是模型区分正负类时的最优临界值。

2.2.3 认证结果的处理方案

因为用户身份认证模型的判断结果不一定正确，存在用户正常使用聊天服务时，持续认证机制判断用户身份认证失败的情况，所以需要使用二次验证确定用户的身份。

当认证机制判断用户认证失败时，聊天服务登出用户并跳转至登陆页面，要求用户再次输入用户名和密码验证身份；当认证机制判断用户认证成功时，聊天服务正常运行。

2.3 用户文本特征更新需求

持续身份认证机制的数据库中包含的用户历史聊天信息的数量有限，通过不断记录用户新产生的聊天信息，模型能够进一步的学习特定用户输出的文本特征，提升分类的查准率和查全率。

同时，用户的文本特征不是一成不变的，其聊天主题、使用的表情包（Emoji 表情）、语气词等会随着热点信息的变化和用户的心情发生改变，所以在基于聊天信息的用户身份认证的过程中，需要更新用户的文本特征，以适应用户的文本风格变化。

当用户认证失败并被强制退出后，再次通过用户名-密码登陆聊天系统时，持续身份认证机制应该检查数据库中导致认证失败的聊天文本。其中发送时间与当前登陆时间接近的聊天文本，应被视作用户认证模型判断错误，实际是由当前用户编写的聊天信息，添加到用户的历史聊天信息中；其它发送时间与当前登陆时间差距较大的聊天文本，会被视作用户认证模型判断正确，实际是由冒用者编写的聊天信息，从数据库中删除。

相比于固定的用户文本特征，此种设计能够较好的保证在用户聊天内容多变、聊天信息长度较短的情况下，准确地对用户进行身份认证，不会因为用户谈论热点信息或者文本风格的改变而导致系统频繁的要求用户输入用户名和密码进行验证。

2.4 非功能性需求

基于聊天信息的持续身份认证机制的非功能性需求是除功能需求以外的特性，是

为了满足该认证机制在实际系统中应用应该具有的需求，主要包括系统的性能需求等。

（1）持续身份认证机制的兼容性

为满足在线聊天系统的用户身份验证需求，基于聊天信息的持续身份认证机制在兼容性上应适用于当前常见的浏览器，同时系统的身份认证环节一般在后端实现，硬件资源相对充足。

（2）机制的吞吐量和响应时间

身份持续认证机制的吞吐量用指持续认证机制承受提交的认证任务的程度，具体为单位时间持续认证机制处理的认证任务的数量，与系统的并发认证请求相关，考虑到在在线聊天场景中，因为聊天身份被冒用导致的财产、信息损失，往往需要冒用者花费 5-10 分钟进行诱导、欺骗，不会在用户身份被冒用的瞬间产生较大影响。

所以本文设计的基于聊天信息的持续身份认证机制应满足，在硬件能力有限的情况下在 5 分钟内处理至少 1000 个用户身份认证请求；单台服务器在同时处理 100 个并发请求的前提下，平均响应时间不超过 15s，最大响应时间不超过 3 分钟。

2.5 本章小结

首先本章节针对实现身份认证机制的两个核心功能模块，用户身份认证模块、用户文本特征更新模块，结合已有的研究成果描述了他们的需求。然后，本章在非功能需求部分，讨论了该机制在实际系统应用中应满足的兼容性、吞吐量和响应时间要求。通过功能性需求和非功能性需求，本章明确了基于聊天信息的持续身份认证机制的设计和实现需要满足的要求。

第三章　持续身份认证机制的概要设计

根据绪论部分的研究现状和需求分析中的描述，开展基于聊天信息的持续认证机制的概要设计，本章将从持续认证机制的架构设计和功能模块展开，逐个论述用户身份认证功能和用户文本特征更新功能两个主要功能模块的技术选型。其中用户身份认证功能是本文设计的持续认证机制的核心，本章将其分为 3.2 节和 3.3 节介绍。

3.1 持续身份认证机制架构设计和功能模块

Baig AF 等人[4]将持续认证定义为在运行期间通过识别用户特征和操作来持续和被动地监控用户。根据定义，持续认证机制应满足能够不断地实时认证用户和认证过程中不打扰用户这两个特征。

本文设计的持续认证机制通过聊天服务在用户聊天过程中，持续转发用户的聊天文本至认证机制，实现对于用户身份的持续认证。认证机制与聊天服务分离，如果当前用户认证通过，持续认证机制不会打扰用户。

3.1.1 持续身份认证机制架构

持续认证机制系统架构图如图 3.1 所示。基于 Spring Boot 的聊天服务在转发、存储用户聊天信息的同时，持续地将用户的聊天信息文本通过 HttpAsyncClient 提交给认证机制。认证机制将基于 Python 后端框架 Fast API 实现，主要功能是通过 Bert-BiGRU-注意力机制组合的认证模型，结合用户的历史聊天文本，判断当前聊天服务转发文本是否由用户本人生成，并通过 requests 发送 HTTP 请求返回认证结果给聊天服务。

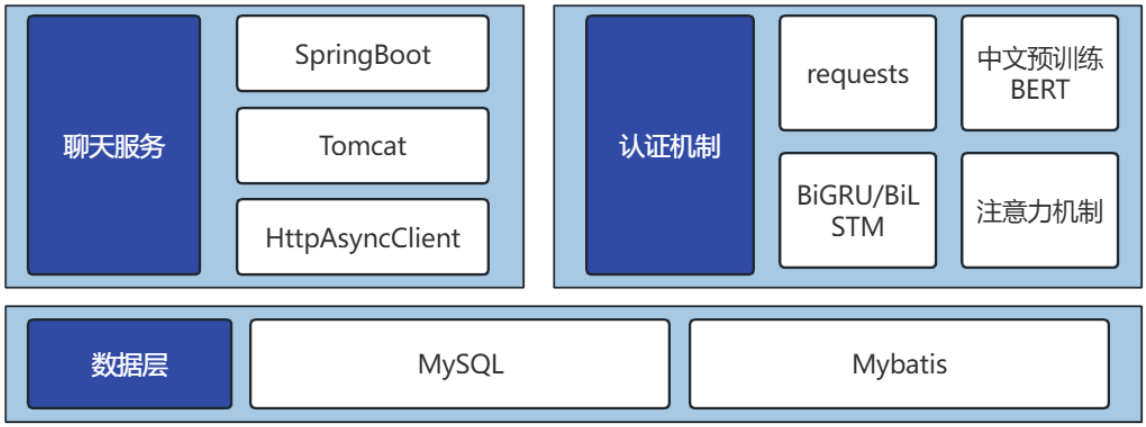


图 3.1 持续认证机制架构图

3.1.2 持续身份认证机制的功能模块

本文设计的基于聊天信息的持续身份认证机制，主要包含聊天信息处理模块、用户身份认证模块和文本特征更新模块。

其中聊天信息处理模块，负责持续地向用户身份认证模块转发用户的聊天文本，实现根据用户聊天信息的持续身份认证。

用户身份认证模块是持续认证机制的核心，通过模型比较用户的历史聊天记录与用户新输入的聊天信息文本，得出新输入的聊天信息属于当前用户的概率。然后，比较计算得到的概率和判断阈值，获得用户身份认证的结果，并将认证结果转发回聊天服务。

用户身份认证失败时，聊天服务会将用户加入黑名单。然后，聊天服务根据黑名单强制退出对应的用户至登陆界面，用户需要输入用户名和密码二次验证身份才能继续使用聊天服务。

在用户身份持续认证成功的情况下，用户输入的聊天信息文本会被正常的添加到数据库中。当用户身份认证失败时，文本特征更新模块会标记导致认证失败的聊天文本，如果用户在短时间内通过二次验证，模块会修改相应聊天文本的标签，更新用户俩是聊天记录；如果用户未能在指定时间内再次登陆，模块会删除标记的聊天文本。

持续认证机制的功能模块图如图 3.2 所示。

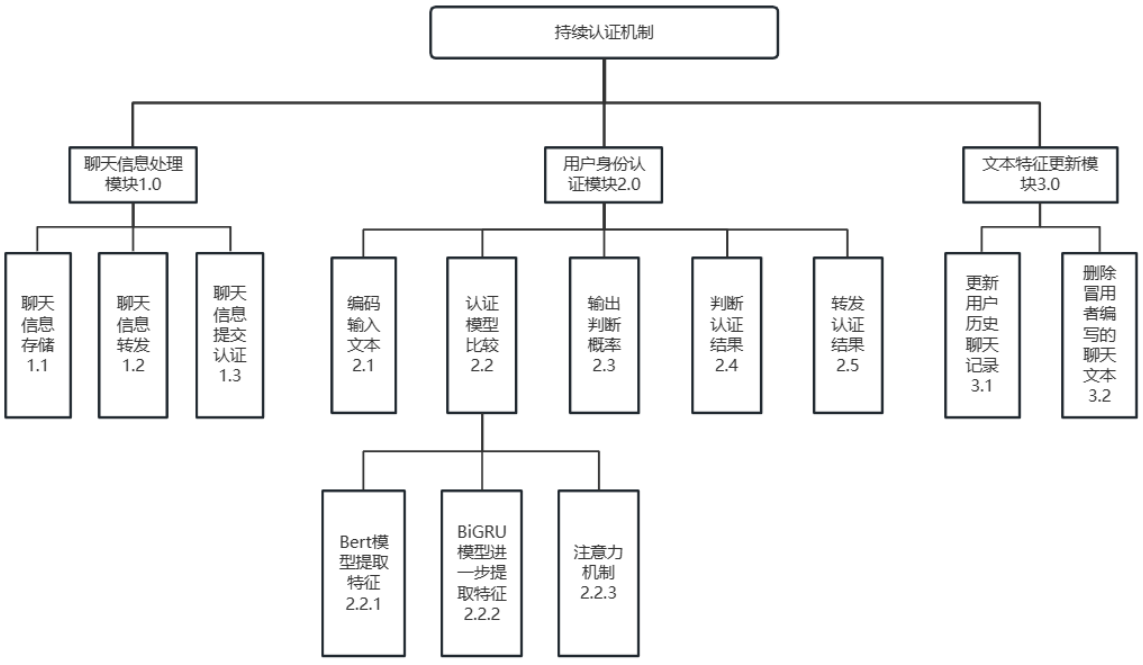


图 3.2 持续身份认证机制功能模块图

3.2 用户身份认证模型设计

用户身份认证模型是判别式模型，输入模型的数据是两段字符串，分别是聊天信息和用户的历史聊天记录，模型通过比较两段字符串之间在用词习惯、造句方式和内容偏好等特征上的相似性，输出聊天信息由用户本人编写的概率。

本节主要内容如下，首先通过比较不同模型组合的认证效果，确定了用户身份认证模型的结构。然后，本节按照认证模型中的数据流向，分别介绍了 BERT 模型、双向 GRU 网络、注意力机制和 softmax 函数的功能和设计方案。

3.2.1 不同模型组合的认证效果比较

由于张翼翔[20]等人的身份识别模型，不需要人工开展复杂的特征工程分析文本数据，适用于短文本、聊天内容多变的场景，故本文考虑使用同类结构实现用户身份认证模型。

具体的模型结构需要通过实验确定，选择传统机器学习算法 SVM 作为基准。其余模型使用张翼翔[20]等人的方案，通过 BERT 模型处理输入的两段字符串，得到聊天信息和历史聊天记录两个字符串的总体特征张量，然后使用不同的模型组合处理特征张量，输出聊天信息由用户本人编写的概率。

实验分别选择双向 LSTM 网络或 GRU 网络作为 BERT 模型的下游模型，降低 BERT 输出特征张量的维度，比较两种网络精炼特征的能力。同时，选择引入注意力机制前后的模型，比较注意力机制作用效果；选择 softmax 函数和 A-softmax 函数分别处理模型的输出结果，比较两种归一化方案的作用效果。

使用 NUS SMS Corpus 数据集中的中文短信数据集 smsCorpus_zh 进行实验，在 5.2.3 节中有训练集和测试集的具体介绍。通过 10 折交叉验证，全面地评价不同模型的认证效果。利用上述方案比较不同模型组合的认证效果，实验结果如表 3.1 所示。

表 3.1 不同模型组合的认证效果实验结果

模型	查准率 (Precision)	查全率 （Recall）	F1 值	AUC	测试样本数量
SVM	0.70	0.81	0.75	无	31465
BERT-BiLSTM(softmax)	0.69	0.69	0.69	0.73	31465
BERT-BiGRU(softmax)	0.75	0.74	0.74	0.79	31465
BERT-BiLSTM-Attention(softmax)	0.79	0.78	0.78	0.82	31465
BERT-BiGRU-Attention(softmax)	0.78	0.77	0.77	0.80	31465
BERT-BiGRU-Attention(A-softmax)	0.79	0.78	0.78	0.82	31465

基于 SVM 算法进行用户身份认证的思路是，针对测试集中的每一个样本，选取相同作者的短信文本作为正样本，选取训练集中其他作者的短信文本作为负样本，训练一个支持向量机分类当前的测试样本。

基于 BERT 的组合模型进行用户身份认证的思路是，构造数据集微调（Fine-tuning）组合模型，使基于 BERT 的组合模型适应基于聊天信息的用户身份认证任务。然后，向组合模型输入两段字符串，分别是用户的聊天信息和历史聊天记录，组合模型输出聊天信息和历史聊天记录由相同作者编写的概率。最后，比较组合模型输出的概率和设定的判断阈值，获得认证结果。

根据表 3.1 展示的实验结果，BERT-BiLSTM-Attention(softmax)、BERT-BiGRU-Attention(softmax)和 BERT-BiGRU-Attention(A-softmax)三个模型组合的认证效果优异。其中使用 softmax 函数处理输出结果的 BERT-BiGRU-Attention 模型，具有结构简单、训练时间短和推理速度快的特点，故本文选用 BERT-BiGRU-Attention(softmax)结构的用户身份认证模型实现基于聊天信息的持续身份认证机制。

3.2.2 基于 BERT 的短文本特征提取

BERT 模型是谷歌研究团队于 2018 年提出的无监督预训练语言模型，其架构是一个多层双向 Transformer 编码器，它使用“掩码语言模型”（MLM）和“下一句预测”（NSP）实现上下文理解[23]，能够捕获丰富的文本特征，适用于文本分类任务。

考虑到中文文本特征提取难度大、聊天信息普遍长度较短、聊天信息用词不规范和训练集数据量有限等因素，本文选用在中文语料上进行预训练的 bert-base-chinese 模型进行最初的文本特征提取，然后将 BERT 模型生成的特征张量输入下游的双向 GRU 网络进行文本深层次信息的提取。

3.2.3 双向 GRU 网络设计

双向门控循环单元网络（Bi-GRU）和双向长短时记忆网络（Bi-LSTM）都是循环神经网络（RNN）的变体，两者都在一定程度上解决了普通 RNN 在长序列训练过程中的梯度消失和梯度爆炸问题。Bi-GRU 和 Bi-LSTM 在 GRU 和 LSTM 的基础上引入双向结构，用于更好地捕获序列数据的双向依赖。

LSTM 单元使用三个门控机制控制信息流，分别是遗忘门、输入门和输出门。其中遗忘门决定丢弃哪些信息，输入门决定更新哪些信息到当前的隐藏状态，输出门决定当前的隐藏状态输出。LSTM 与 GRU 相比更复杂，能够有效地捕捉长期依赖。

GRU 单元是 LSTM 单元的精简，由两种门控机制组成，分别是重置门和更新门。其中重置门决定上一步隐藏状态的输出对当前的影响，更新门决定新信息、旧信息的混合比例和当前隐藏状态的更新。与 LSTM 单元相比，GRU 单元具有更少的参数、训

练速度更快，适用于短文本或序列较短的任务。GRU 单元结构如图 3.3 所示。

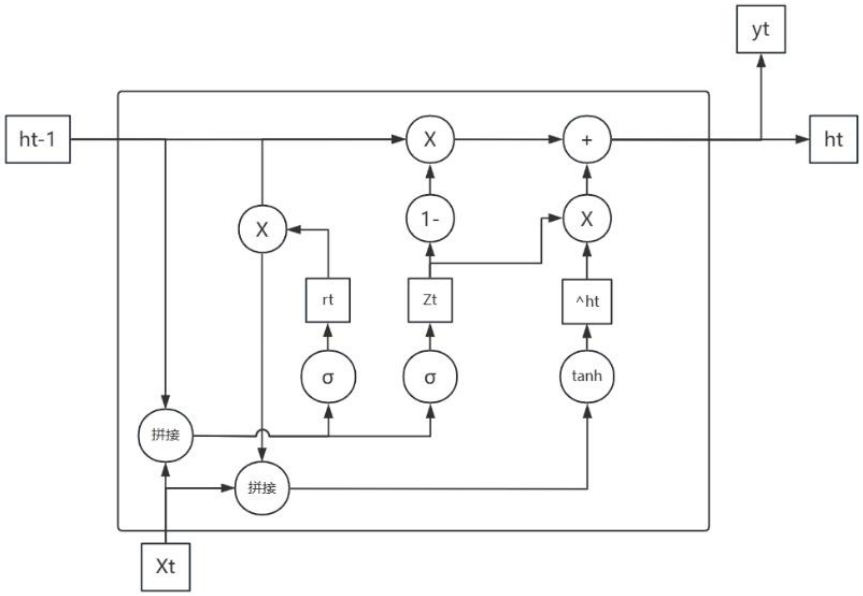


图 3.3 GRU 单元结构图

由于 LSTM 和 GRU 网络中信息的传递是单向的，无法有效地捕获聊天文本的上下文关联信息，所以使用 Bi-LSTM 和 Bi-GRU。Bi-LSTM 和 Bi-GRU 在标准的 LSTM 和 GRU 的基础上，使用两组 LSTM 和 GRU 单元，一组用于从前到后地处理序列，另一组用于从后到前地处理序列，能够更充分地捕获语句序列的文本特征，最终的输出由前向和后向的 LSTM 或 GRU 单元拼接得到。

本文使用双向 GRU 网络作为 BERT 的下游模型，进一步提取文本的深层次信息，然后引入注意力机制将提取到的特征向量加以优化。Bi-GRU 模型结构如图 3.4 所示。

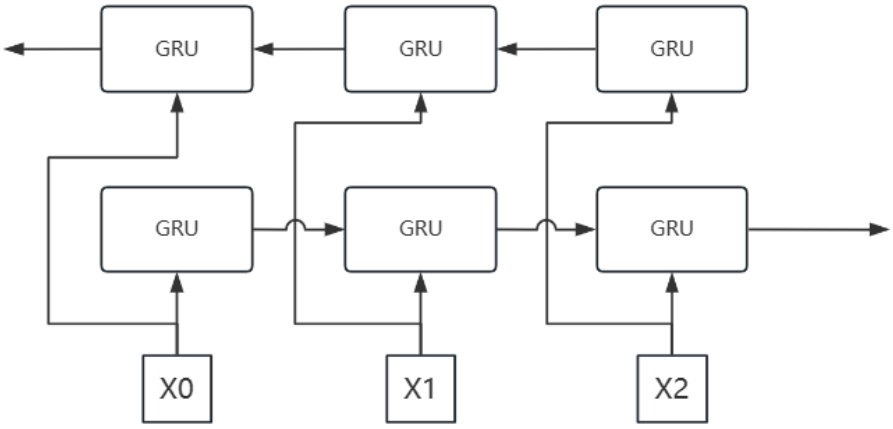


图 3.4 BiGRU 模型结构图

3.2.4 注意力机制

在处理复杂的输入数据时，数据的某些部分对输出的贡献比其他部分更重要，基

于此 Mnih 等人[24]通过强化学习实现了任务驱动的注意力机制。注意力机制在自然语言处理领域有广泛的应用，能够赋予文本中词语不同的权重，进而更准确地理解序列的语义。

本文设计的用户认证模型在注意力机制层对上层模型输出的特征向量中的重要内容分配权重，并将处理后的特征向量输入 softmax 函数处理，得出当前文本由用户本人编写的概率，实现用户认证。注意力机制的实现流程如下：

1.首先将 Bi-GRU 每个时间步的隐藏状态通过线性变换映射为注意力得分，线性变换的公式如(3.1)所示。

$$\mathrm{e}_t = \mathrm{L}inear(h_t) = W \cdot h_t + b \tag{3.1}$$

式(3.1)中h_t表示 Bi-GRU 输出中第 t 个时间步的隐藏状态，W表示线性层的权重，b表示线性层的偏置，e_t表示未归一化的注意力得分。

2.使用 Softmax 函数归一化注意力得分，使其在时间维度（dim=1）上变为概率分布，变换公式如(3.2)所示。

$$\propto_t = \frac{\exp\,(e_t)}{\sum_{k=1}^{seq\_len} \exp\,(e_k)} \tag{3.2}$$

式(3.2)中$\propto_t$表示第 t 个时间步的注意力权重，即第 t 个时间步对于整体序列表示的贡献。

3.将注意力权重应用到 GRU 输出的每个时间步上，如公式(3.3)所示。

$$h_t^{weighted} = \propto_t \cdot h_t \tag{3.3}$$

式(3.3)中$h_t^{weighted}$表示第 t 个时间步加权后的特征表示。

4.对所有时间步的加权特征进行求和，得到整个序列的全局表示，如公式(3.4)所示。

$$H^{final} = \sum_{t=1}^{seq\_len} \propto_t \cdot h_t \tag{3.4}$$

式(3.4)中H^{final}是序列的全局表示，这种表示是文本序列整体信息的浓缩形式。将H^{final}作为模型全连接层的输入，最后通过 softmax 函数处理模型全连接层的输出，得到当前文本由用户本人编写的概率，实现用户认证。

3.2.5 softmax 函数

softmax 函数是用于多分类问题的激活函数，它能够将任意长度为 K 的实向量中的每个分量转换为概率分布，转换后所有分量的和为 1。函数的定义如公式(3.5)所示。

$$\mathrm{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.5}$$

Liu 等人[25]提出的 A-softmax 函数是 softmax 函数的改进版本，函数的核心思想是在特征空间中引入角度间隔(angular margin)，使同类样本的特征向量在球面空间中的分布更紧凑，使不同类别样本之间的决策边界更明显。

根据 3.2.1 节中的表 3.1 的实验结果，在同样使用 BERT-BiGRU-Attention 结构的模型时，使用 softmax 函数和 A-softmax 函数处理模型输出结果的方案在查准率、查全率和 F1 值上接近，但使用 softmax 函数的方案模型结构简单、训练时间短。

因此，本文设计的用户身份认证模型使用 softmax 函数处理 BERT-BiGRU-Attention 模型全连接层输出的原始向量（logits）。

3.3 判断阈值和认证结果处理方案设计

根据 2.1.1 节用户身份认证功能的需求分析，用户身份认证模块的设计包括用户身份认证模型设计、判断阈值设计和认证结果的处理方案设计三个部分。其中用户身份认证模型设计已经在 3.2 节中说明，本节将介绍判断阈值设计和认证结果的处理方案设计两个部分。

3.3.1 判断阈值设计

本文设计的用户身份认证模型，能够比较用户当前的聊天文本和数据库存储的用户历史聊天文本，判断当前的聊天文本是否由用户本人编写。其中，用户身份认证模型的输出结果经过 softmax 函数处理，输出向量中的两个分量分别表示聊天文本由用户本人编写的概率和聊天文本不是用户本人编写的概率。

但 softmax 函数处理后的模型输出结果，并不是聊天文本由用户本人编写的真实概率，而是用户身份认证模型判断聊天文本由用户本人编写的置信度的归一化处理。因此，不能简单地选择 0.5 作为模型输出结果的判断阈值，而是应该基于 NUS SMS Corpus 数据集开展模型测试，并根据测试结果绘制 ROC 曲线，选择 ROC 曲线上 TPR 值与 FPR 值的差最大的点的索引值作为判断聊天文本是否由用户本人编写的阈值。

对比 softmax 函数的输出结果和判断阈值，当新输入的聊天信息属于用户的概率低于判断阈值时，持续身份认证机制判断用户认证失败；当新输入的聊天信息属于用户的概率高于判断阈值时，持续身份认证机制判断用户认证通过。

3.3.2 认证结果处理方案设计

持续身份认证机制完成用户身份认证后向聊天服务发送认证结果，如果用户认证成功，聊天服务正常运行；如果用户认证失败，聊天服务将用户添加至黑名单，准备强制退出用户，并标记导致用户认证失败的聊天文本。

用户切换聊天页面、刷新聊天页面或发送聊天消息时，持续身份认证机制会检查用户是否在黑名单中，如果用户在黑名单中，聊天服务的前端会清除用户的身份令牌（Token）强制退出用户，同时跳转页面至登陆页面，要求用户再次输入用户名密码验

证身份。

3.4 用户文本特征更新机制设计

用户文本特征更新机制旨在动态地积累和更新用户的文本特征，提升用户身份认证模型的分类能力。

3.4.1 存储用户聊天信息的数据库的设计

本文设计的基于聊天信息的持续身份认证机制总共包含四张数据表，分别是用户信息表、用户聊天对象关系表、用户聊天信息记录表和黑名单。其中黑名单的结构简单，仅用于存储需要被强制退出的用户的识别码，此处不做介绍。

用户信息表用于存储用户信息，包含的属性有用户身份识别码、用户真实姓名、用户名、密码、性别和电话号码。

用户聊天对象关系表用于存储用户之间的聊天关系，包含的属性有用户身份识别码、聊天对象的用户身份识别码和最后一条聊天信息的发送时间。

用户聊天信息记录表用于存储用户的历史聊天信息，包含的属性有聊天信息的编号、用户身份识别码、聊天对象的用户身份识别码、聊天信息的发送时间、聊天信息文本、标志位。根据上述三张数据表绘制的数据库的 E-R 图如图 3.3 所示。

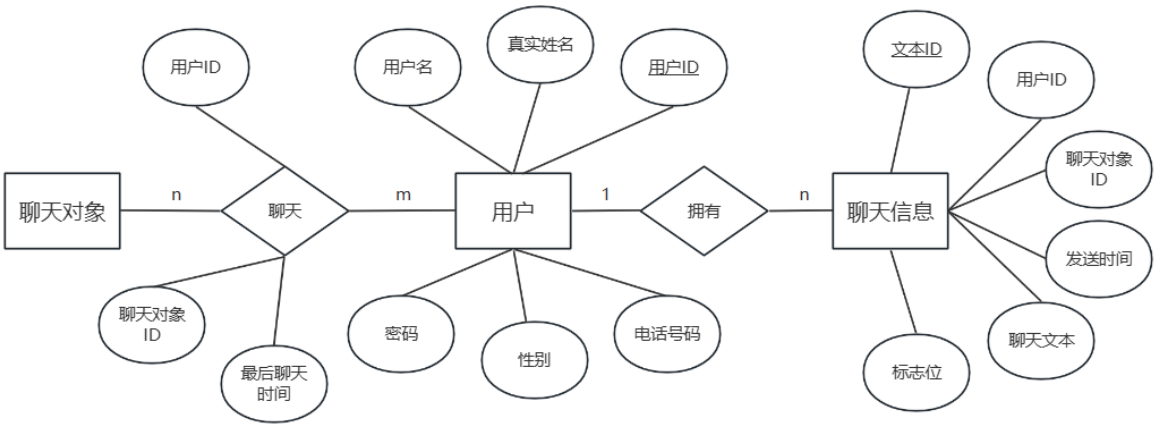


图 3.5 数据库的 E-R 图

3.4.2 文本特征更新机制

用户的文本特征受到用户的阅历、爱好和习惯等诸多个人因素影响，这些影响因素通常会随着时间发生改变，所以在基于聊天信息的用户身份认证的过程中，需要更新用户的文本特征，以适应用户的文本风格变化。

文本特征更新机制的具体实现方案如下，如果用户在聊天过程中持续认证成功，

用户输入的聊天信息的标志位全部设置为 1，表示这些聊天信息通过认证，由用户本人编写。如果用户认证失败，未通过认证的聊天信息的标志位会被设置为 0，表示这条聊天信息有可能是由冒用者编写。

当用户认证失败并被强制退出后，再次通过用户名-密码验证登陆聊天系统时，文本特征更新机制会检查数据库中标志位为 0 的文本。其中发送时间与当前登陆时间相差 3 分钟之内的聊天信息，会被视作用户本人编写的聊天信息，修改这条聊天信息的标志位为 1，将聊天信息添加到当前用户的历史聊天记录中。发送时间与当前登陆时间相差 3 分钟以上的聊天信息，会被视作由冒用者编写的聊天信息，从数据库中删除。

3.5 本章小结

本章介绍了基于聊天信息的持续身份认证机制的概要设计，先通过持续认证机制的架构图和功能模块图介绍了本文设计的持续身份认证机制的总体架构和主要的功能模块，然后从用户身份认证模型、模型输出结果的判断和处理、用户文本特征更新机制的设计详细介绍了持续身份认证机制各个主要功能模块的技术选型和实现方案。

第四章　持续身份认证机制的详细设计与实现

根据概要设计章节的持续身份认证机制架构和功能模块，本章将从开发环境、持续认证机制整体的设计与实现、用户身份认证功能的设计与实现和用户文本特征更新功能的设计与实现四个方面来开展。其中用户认证功能是持续认证机制的核心，本章将其分为 4.3 节和 4.4 节介绍，分别说明用户身份认证模型、判断阈值和认证结果处理方案的设计与实现。

4.1 开发环境

本文设计的基于聊天信息的持续身份认证机制的开发语言主要涉及 Java 和 Python 两门语言，在聊天服务前端的实现过程中，简单使用了 HTML、CSS、JavaScript 三门语言。

在开发过程中使用 Vue2 框架和 Spring Boot 框架分别搭建聊天服务的前后端，主要使用 WebStorm 集成开发工具和 Vue CLI 等工具进行聊天服务的前端开发；使用 Intellij IDEA 集成开发工具进行聊天服务的后端开发，使用持久层框架 Mybatis 实现聊天数据的存储和用户文本特征的更新；使用 Python 后端框架 Fast API 实现身份认证机制的后端，使用 PyCharm 集成开发工具进行后端代码的开发。

在数据层，本文设计的持续认证机制使用 MySQL 作为数据库管理工具，与后端框架 Spring Boot 具有良好的适配。

在机器学习框架的选择上，本文使用 Transformers 库中的 BERT 模型和 BERT 分词器构建组合模型，使用 PyTorch 框架实现结构为 BERT-BiGRU-Attention(softmax)的用户认证模型。

4.2 持续身份认证机制整体的设计与实现

根据 3.1 中的描述，本文设计的持续身份认证机制的主要功能与聊天服务和认证机制两个部分有关。其中，聊天服务用于模拟在线聊天平台，负责用户登陆、用户身份验证、存储用户聊天信息和转发认证信息至认证机制；认证机制负责接收认证信息、编码认证信息、运行身份认证模型、判断认证结果和转发认证结果。

具体而言，使用 Vue2 和 Spring Boot 框架实现聊天服务的前后端，用户登陆聊天服务后，聊天服务后端会向聊天服务前端发送 Token 作为用户的登陆凭证，Token 能够在聊天服务的后端中被解析为用户的识别码（uid）。其中聊天服务后端通过

HttpAsyncClient 构建异步的 HTTP 请求，将用户的身份识别码（uid）、历史聊天记录和新发送的聊天信息转发至认证机制后端。

使用 Fast API 框架实现认证机制后端，认证机制首先编码接收到的认证信息，然后使用身份认证模型计算用户新发送的聊天信息由用户本人编写的概率，并比较概率和判断阈值得到认证结果，最后通过 requests 构建 HTTP 请求，将用户的身份识别码（uid）、判断结果和用户新发送的聊天信息转发至聊天服务后端。

聊天服务接收认证结果，如果用户认证失败，将用户加入黑名单中，标记导致认证失败的聊天文本，然后聊天服务根据黑名单强制退出对应用户至登陆界面；如果用户认证成功，不进行任何操作。

持续身份认证机制的整体流程如图 4.1 所示。

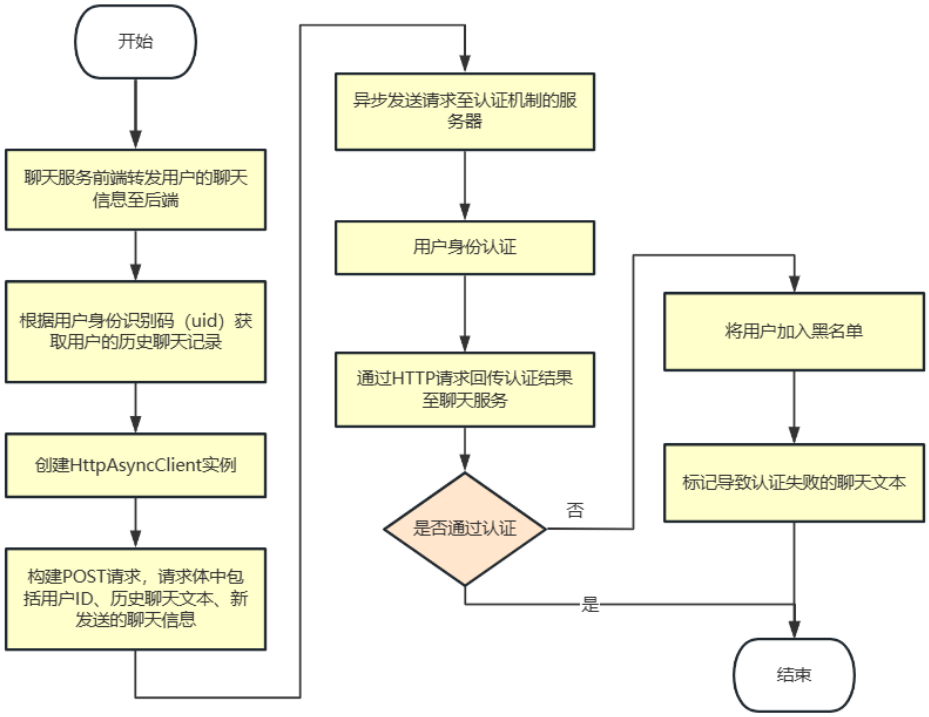


图 4.1 持续身份认证机制的流程图

根据上述流程，基于用户输入的每一条聊天信息认证用户身份，实现基于聊天信息的持续身份认证机制。

4.3 用户身份认证模型的设计与实现

用户身份认证功能接收聊天服务转发的信息后，将聊天信息和历史聊天文本两段字符串输入用户身份认证模型，用户身份认证模型输出两段字符串由相同作者编写的概率。

本节首先介绍如何利用 Transformers 库中的 BERT 模型、BERT 分词器和 PyTorch 框架实现用户身份认证模型。

在使用模型之前，需要构造数据集训练模型，使模型适应基于聊天信息的用户身份认证任务。所以在模型搭建完成后，本节介绍了如何基于 NUS SMS Corpus 中文短信数据集构建微调模型的数据，并说明用户身份认证模型的训练过程，展示模型训练后的认证效果。

用户身份认证模型处理聊天信息的流程图如图 4.2 所示。

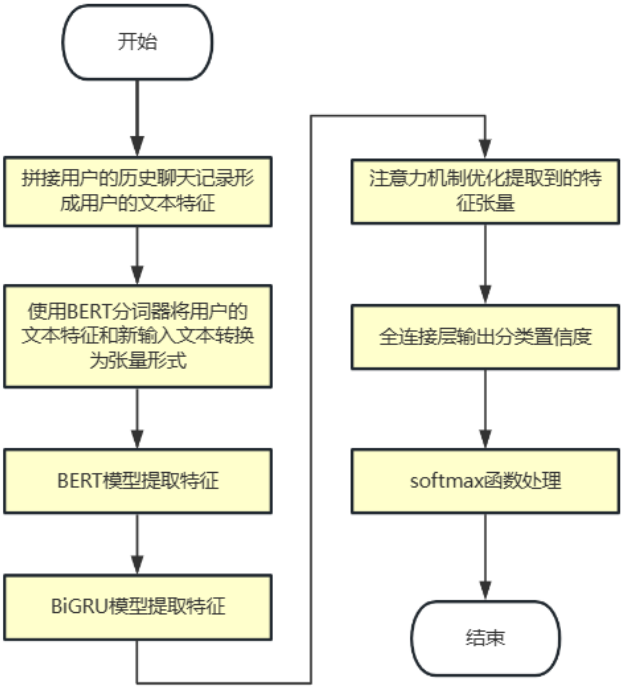


图 4.2 用户身份认证模型处理聊天信息的流程图

4.3.1 用户身份认证模型的实现

根据 3.2 用户身份认证模型设计，用户身份认证模型的结构为 BERT-BiGRU-Attention(softmax)，下文将按照模型中的数据流向，分别介绍 BERT 模型、双向 GRU 网络、注意力机制和 softmax 函数的实现。

首先介绍 BERT 模型，因为本文设计的持续认证机制基于中文聊天文本开展认证，所以使用在中文语料上进行预训练的 bert-base-chinese 模型。BERT 模型作为用户身份认证模型的基础，负责把输入的聊天信息文本和历史聊天文本中的每个字符转换成 768 维的特征向量，完成文本特征提取。

双向 GRU 网络的输入是 768 维的特征张量，输出是两个 128 维的特征张量，使用 PyTorch 中的 GRU 模块实现。双向 GRU 网络接收 BERT 模型的输出，精简 BERT 模

型提取到的特征，将 BERT 模型输出的 768 维的特征张量降低为 256 维的特征张量。

注意力机制的输入是 256 维的特征张量，输出是一个 256 维的特征向量，表示输入的两段文本的权重，使用 PyTorch 中的线性层和 softmax 函数实现。注意力机制接收双向 GRU 网络的输出，通过线性层计算每个字符的注意力权重，使用 softmax 函数归一化权重，最后通过加权求和得到一个 256 维向量，表示两段输入文本的整体特征向量。

然后，使用线性层处理注意力机制输出的特征向量，输出两个值，分别表示两段输入文本的作者不同的置信度和两段输入文本作者相同的置信度。最后，通过 softmax 函数归一化处理两个置信度，得到两段文本作者不同的概率和两段文本作者相同的概率。

选择两段文本作者相同的概率作为模型的输出结果，用于后续判断。用户身份认证模型的结构如图 4.3 所示。

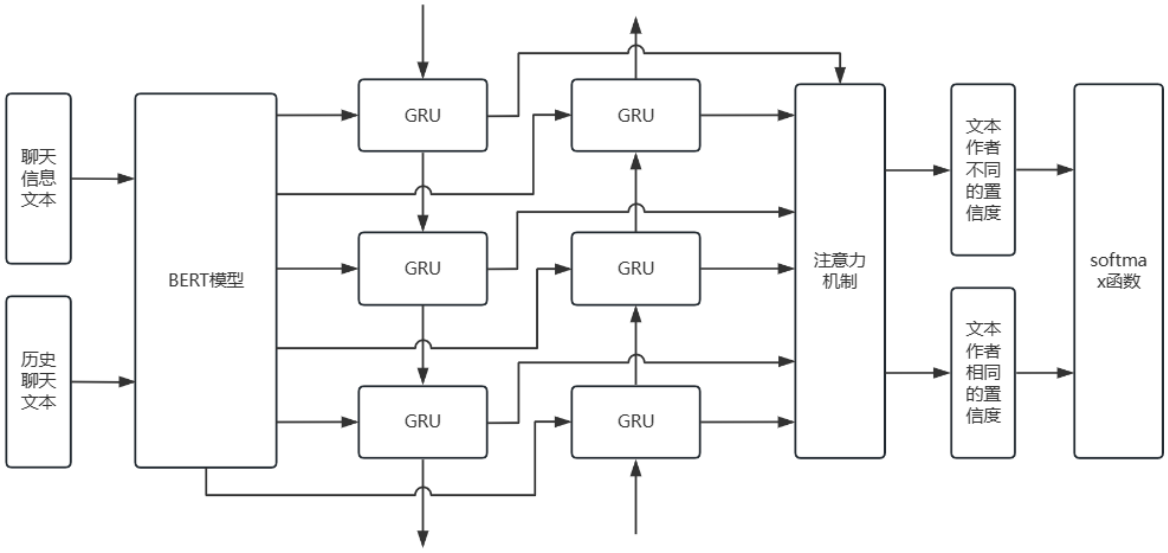


图 4.3 用户身份认证模型的结构示意图

4.3.2 构建数据集微调模型

NUS SMS Corpus 中文短信数据集中仅包含文本作者、短信文本等信息，没有正负样本的区别。因此考虑将用户的历史聊天记录文本和用户的一条聊天文本的组合视作正样本，将其他用户的历史聊天记录文本和用户的一条聊天文本的组合视作负样本，让用户身份认证模型学习同一用户不同聊天文本间的关联和不同用户聊天文本间的差异，实现基于聊天信息的用户认证功能。

虽然基于聊天信息的用户认证任务是一个二分类任务，即判断当前文本是否由用户本人编写，但是任务的正负样本并不平衡，因为一个用户可能被多个冒用者冒充。因此在构建训练集时，需要适当增加负样本的数量，让用户身份认证模型能够充分学习到不同用户的文本特征之间的差异，更好地识别用户和冒用者。

基于上述思路，构建微调模型的数据集的算法如下所示。

模型微调数据集构建算法

输入：

train_set：基于 NUS_SMS_Corpus 数据集随机抽取构造的训练集

negative_sample_count：微调时每条正样本对应的负样本数量

输出：

training_samples：模型微调数据集

1.获取训练集中的所有用户 all_users

2.初始化 training_samples

3.对于 all_users 中的每个用户 user

a.获取 train_set 中与 user 对应的所有历史记录 user_rows

b.获取 user_history 中 user 的历史记录 user_history_text

c.对于 user_rows 中的每一行 row:

i.构造一个正样本：

-文本特征：user_history_text

-判断文本：row 的 content 属性

-标签：1

并添加到 training_samples

ii.初始化空列表 negative_samples

iii.从 all_users 中随机选择 negative_sample_count 个用户作为负样本用户 negtive_users，选择时排除当前用户 user

iv.对于每个负样本用户 neg_user：

-从 user_history 中获取 neg_user 的历史记录 neg_history

-构造一个负样本：

-文本特征：neg_history

-判断文本：row 的 content 属性

-标签：0

并添加到 negative_samples

v.将负样本列表 negative_samples 添加到 training_samples

4.将 training_samples 转换为 DataFrame 并输出

使用上述数据集微调模型时，首先通过前向传播计算模型预测结果和真实标签之

间的差值获得损失值，然后执行反向传播，设置学习率为 0.00001，根据损失值计算参数梯度并更新参数，微调模型。

用户认证模型训练过程中训练集和测试集的损失随着训练轮数的增加而改变的趋势如图 4.4 所示。图像显示，随着训练轮数（epoch）的增加，训练集损失逐渐下降，测试集损失逐渐提高，因此用户认证模型仅需训练一轮就能收敛。

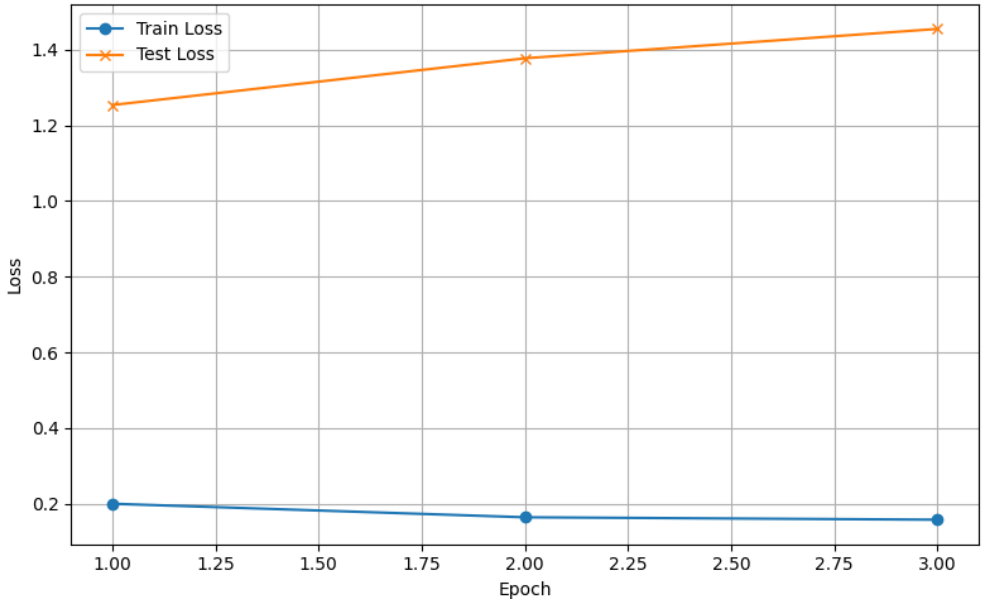


图 4.4 用户认证模型训练过程中训练集和测试集的损失变化情况

模型微调后，开展交叉验证测试模型性能，根据测试结果绘制的 ROC 曲线的平均 AUC 值为 0.80，微调后的用户身份认证模型能够基于聊天信息认证用户身份。模型微调前后的测试结果的 ROC 曲线见 5.3.1 节中的图 5.4。

同时，微调后的身份认证模型的测试结果在查准率、查全率和 AUC 值等数值上表现良好，具体数值见 5.3.1 节中的表 5.5。

4.4 判断阈值和认证结果处理方案的设计与实现

经过 softmax 函数处理的用户认证模型的输出结果，并不是聊天文本由用户本人编写的真实概率，而是用户身份认证模型判断聊天文本由用户本人编写的置信度的归一化处理。

因此不能简单地选择 0.5 作为模型输出结果的判断阈值，需要使用 5.2.3 节用户身份认证功能测试中构造的测试集开展测试，根据测试结果选定用户身份认证的判断阈值。

根据测试结果绘制的 ROC 曲线在 5.3.1 节中的图 5.4 中已经展示，选择 ROC 曲线上 TPR 值与 FPR 值的差最大的点的索引值 0.2063 作为模型输出结果的判断阈值。

用户认证模型根据聊天服务传入的数据，输出新发送的聊天信息由用户本人编写的概率。将模型输出的概率和判断阈值对比，当模型输出的概率大于判断阈值时，认为新发送的聊天信息由用户本人编写，用户身份认证成功；否则认为新发送的聊天信息由冒用者编写，用户认证失败。

完成模型输出结果的判断后，将认证结果转发回聊天服务。如果用户认证失败，则将用户的身份识别码（uid）添加到黑名单中，同时在数据库中将导致用户认证失败的聊天信息标记。

基于上述思路，模型输出结果的判断和处理的流程如图 4.5 所示。

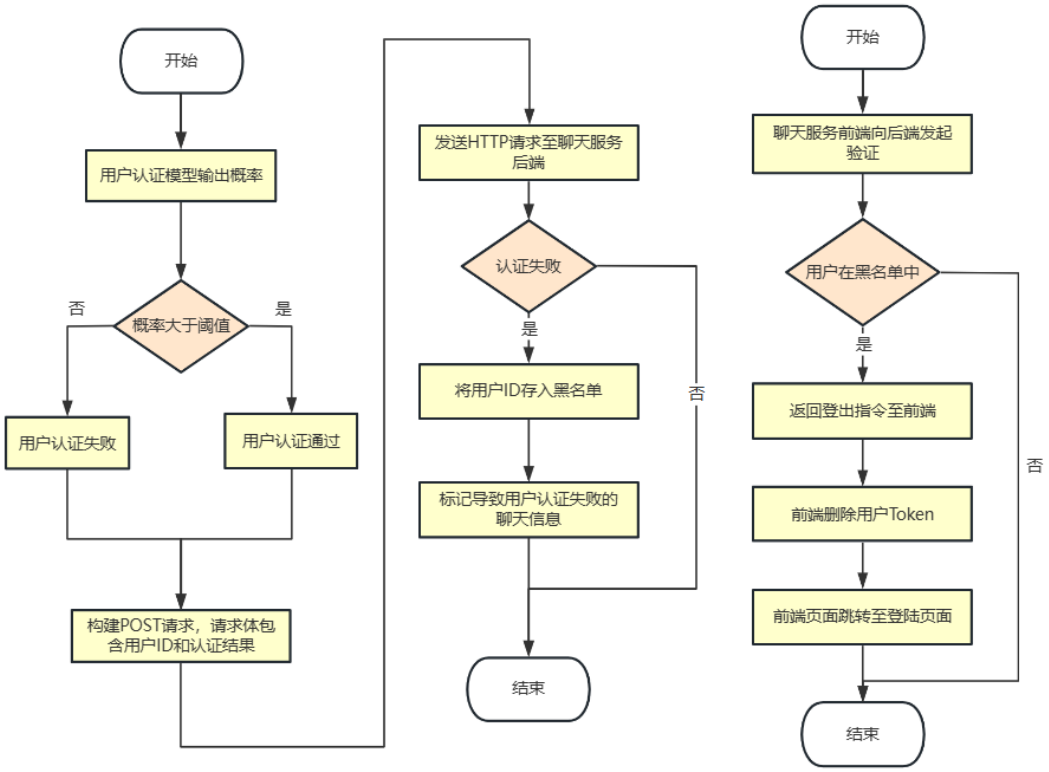


图 4.5 模型输出结果的判断和处理功能的流程图

聊天服务的前端在用户刷新页面、切换页面和发送聊天信息时，会向后端发起验证，检查用户是否在黑名单中。当用户在黑名单中，后端会向前端返回指令，删除用户的登陆令牌（Token）强制退出当前用户，并跳转页面至登陆页面，要求用户再次输入用户名密码验证身份。

4.5 用户文本特征更新机制

用户文本特征更新机制旨在动态地积累和更新用户的文本特征，提升用户身份认证模型的分类能力。本节将具体介绍存储用户聊天信息的数据库和用户文本特征更新

机制的实现方案。

4.5.1 存储用户聊天信息的数据库的设计与实现

根据 3.4.1 节中的 E-R 图设计用户信息表、用户聊天对象关系表和用户聊天信息记录表。由于黑名单仅存储认证失败的用户的身份识别码（uid），在实际应用中，能够使用 Redis 缓存等其他手段实现，故此处不做介绍。

用户信息表用于存储用户信息，结构如表 4.1 所示，包含的属性有用户身份识别码（id）、用户真实姓名、用户名、密码、性别和电话号码。其中用户身份识别码（id）是主键，能够唯一标识用户身份；用户密码经过 MD5 加密后才保存在数据表中，能够有效的防止数据库泄露导致用户的同用户名同密码的其他账号遭受攻击。

表 4.1　用户信息表

字段名称	字段说明	数据类型	字段大小	运行空
id	用户身份识别码	bigint		否
real_name	用户真实姓名	varchar	180	
name	用户名称	varchar	180	否
password	密码	varchar	255	否
gender	性别	varchar	10	否
phoneNum	电话号码	int		否

用户聊天对象关系表用于存储用户之间的聊天关系，包含的属性有用户身份识别码（uid）、聊天对象的用户身份识别码和两者间最后一条聊天信息的发送时间。因为聊天关系是用户之间的多对多关系，所以这张数据表没有主键。用户聊天对象关系表主要用于搭建聊天服务，不存储持续身份认证机制所需数据。

用户聊天对象关系表的结构如表 4.2 所示。

表 4.2　用户聊天对象关系表

字段名称	字段说明	数据类型	字段大小	运行空
uid	用户身份识别码	bigint		否
chatPartner	聊天对象身份识别码	bigint		否
time	最后一条聊天记录发送时间	datetime		否

用户聊天信息记录表用于存储用户的历史聊天信息，包含的属性有聊天信息的编号、用户身份识别码、聊天对象的用户身份识别码、聊天信息的发送时间、聊天信息文本、标志位。其中聊天信息的编号（content_id）是主键，能够唯一标识一条聊天信息。

标志位用于区分聊天文本，标志位为 1 表示聊天文本通过模型认证，可以视作用户文本特征的组成部分；标志位为 0 表示聊天文本导致用户认证失败，需要用户文本特征更新机制进一步处理。用户聊天信息记录表的结构如表 4.3 所示。

表 4.3　用户聊天信息记录表

字段名称	字段说明	数据类型	字段大小	运行空
content_id	聊天信息编号	bigint		否
uid	用户身份识别码	bigint		否
chatPartner_id	聊天对象用户 ID	bigint		否
time	聊天信息发送时间	datetime		否
content	聊天信息文本	text		否
tag	标志位	int		否

4.5.2 文本特征更新机制的设计与实现

在聊天过程中用户的话题会发生变化、用词习惯会受网络热门用语影响，即用户的文本特征会随着时间发生改变，所以需要相应的更新机制适应用户的文本风格变化。

当用户的文本风格发生明显的变化时，基于聊天信息的持续身份认证机制通常会判定用户认证失败。因此实现文本特征更新机制重点在于区分用户编写的聊天信息和冒用者编写的信息。文本特征更新机制的流程如图 4.6 所示。

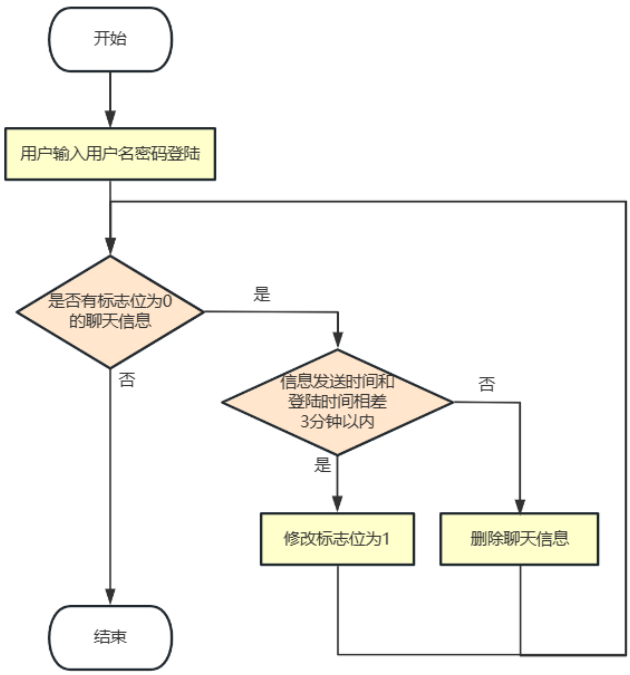


图 4.6 文本特征更新机制的流程图

在聊天过程中，如果是用户本人因为文本风格变化导致身份认证失败，通常会在较短的时间内通过用户名密码验证，再次登陆；反之，如果是冒用者触发的认证失败，在用户的用户名和密码信息安全的情况下，冒用者通常不能在短时间内再次操控用户账号。

因此文本特征更新机制的实现方案如下，用户登陆时会检查数据库中用户的聊天信息，如果用户有标志位为 0 的聊天信息，则对比此次用户登陆时间和标志位为 0 的聊天信息的发送时间。其中两个时间相差大于 3 分钟的聊天信息被视作冒用者编写的聊天信息，从数据库中删除；两个时间相差小于 3 分钟的聊天信息被视作用户本人编写的聊天信息，修改标志位为 1。重复执行此过程直至用户的历史聊天记录中没有标志位为 0 的聊天信息，完成用户的文本特征更新。

4.6 本章小结

本章首先介绍了使用的语言、开发工具等，然后介绍了持续认证机制整体的设计与实现，使读者了解本文设计的基于聊天信息的持续身份认证机制的主要流程和实现方案。然后，本章分别介绍了用户身份认证功能和文本特征更新功能两个主要的功能模块的设计与实现，补充说明了本文设计的持续认证机制的实现细节。

第五章　持续身份认证机制测试

本章将使用 NUS SMS Corpus 中的中文短信数据集和一些具有代表性的测试用例，针对第四章中实现的基于聊天信息的持续身份认证机制开展测试，全面地评价本文设计的持续身份认证机制。

5.1 实验和测试设计

按照需求分析开展实验和测试，测试本文设计的持续认证机制是否满足功能性需求和非功能性需求。

5.1.1 功能性测试设计

根据需求分析中的描述，系统的功能性需求包括用户身份认证需求和用户文本特征更新需求。其中，用户身份认证功能包含用户身份认证模型、判断阈值和认证结果的处理三个功能。

由于模型的认证效果与判断阈值的设定有关。因此，本章的功能性测试按照认证模型及判断阈值测试、认证结果处理功能测试和用户文本特征更新功能测试三个部分开展。

认证模型及判断阈值测试的目标是，测试本文设计的模型和判断阈值能否根据聊天信息认证用户的身份。使用 ROC 曲线的 AUC 值（Area Under Curve）评价认证模型的身份认证能力，使用查准率、查全率和 F1 值综合评价认证模型和判断阈值的身份认证能力。基于公开数据集 NUS SMS Corpus 的中文短信库构造测试数据，开展 10 折交叉验证，测试数据的具体情况在 5.2.3 节中说明。

认证结果处理功能测试的目标是，验证本文设计的持续认证机制能否根据认证结果，正确的触发二次验证，保证正常用户的使用权限和拦截非法用户。测试方案是使用涵盖不同情况的测试用例，观察认证结果处理功能是否正常。

用户文本特征更新功能测试的目标是，验证认证机制能否根据用户登陆时间与聊天信息发送时间之间的时间差，保留正常用户的聊天信息，删除冒用者编写的聊天信息。测试方案是使用涵盖不同情况的测试用例，观察文本特征更新功能是否正常。

5.1.2 非功能性测试设计

根据需求分析中的描述，系统的非功能性测试包括兼容性测试、吞吐量测试和响应时间测试。

兼容性测试的目标是，验证基于聊天信息的持续身份认证机制在常见的浏览器上能否正常运行。测试方案是使用不同的浏览器运行本文设计的持续认证机制，观察机制的各项功能是否正常。

吞吐量测试的测试目标是，评估本文设计的用户身份认证服务在单位时间内处理请求的能力，验证用户身份认证服务的性能和稳定性。测试方案是使用 Locust 框架模拟并发身份认证请求，观察不同并发数量、不同测试时长下，认证服务的吞吐量和成功率。

响应时间测试的测试目标是，评估本文设计的用户身份认证服务在不同并发数量下处理请求所需的时间，验证本文设计的持续认证机制能否快速地认证用户身份和处理冒用者。测试方案是使用 Locust 框架模拟并发身份认证请求，观察不同并发数量、不同测试时长下，认证服务的平均响应时间、最大响应时间和最小响应时间。

通过 Locust 构建 POST 请求向认证服务后端的 predict 接口发送模拟数据，模拟并发身份认证请求。每个模拟数据包含用户身份识别码（id）、用户历史聊天信息和聊天文本，与实际工作环境下聊天服务传递给认证服务的数据一致。

5.2 测试环境和测试数据

本节主要介绍测试环境和测试数据，其中测试环境部分介绍了测试环境的硬件配置；测试数据部分介绍了公开数据集（NUS SMS Corpus）中包含的属性、数据集的分布特点等内容，为功能性测试和非功能性测试的开展做准备。

5.2.1 测试环境与测试工具

测试环境：

测试环境如表 5.1 所示。

表 5.1 测试环境

配置项	详细信息
操作系统	Windows10 家庭版
CPU	Intel(R)Core(TM) i5-9300H CPU @ 2.40GHz
内存	16GB
GPU	NVIDIA GTX 1650

测试工具：

使用 Python3.9 和 PyTorch2.5.1 框架进行模型训练及测试，使用 Locust 框架进行性能测试，使用集成开发环境 PyCharm 进行测试代码编写。

5.2.2 公开数据集的基本情况

测试使用的公开数据集为 NUS SMS Corpus 中的中文短信库，使用的版本为 2015.03.09 版。该数据集由新加坡国立大学研究者 Tao Chen 收集，包含来自 594 名作者的 31465 条数据[21]。

每条数据包括文本发送者（sender）、接收者（receiver）、发送时间（send_time）、收集时间（collect_time）、收集方式（collect_method）、短信文本（content）、是否是本地人（native）、短信作者的国家（country）、年龄（age）、短信文本生成方式（input_method，包含拼音、笔画和其他）、使用手机的经验（experience）、每天发送短信的频率（frequency）、性别（gender）、语言（lang）、城市（city）等属性。数据集的概况如表 5.2 所示。

表 5.2 数据集的样本概况

数据总量	短信文本数量	短文本的作者数量	每个作者平均文本数量	作者对应文本数量的最大值	作者对应文本数量的众数
31465	31465	594	52.97	2104	20

该数据集中每个短信作者的短信文本数据量的分布情况如图 5.1 所示，数据集中总共包含 594 位短信作者，其中有 22 位短信作者总共发布了 1-9 条短信文本，有 115 位短信作者总共发布了 10-19 条短信文本，有 324 位短信作者总共发布了 20-29 条短信文本，有 53 位短信作者总共发布了 30-49 条短信文本，有 31 位短信作者总共发布了 50-99 条短信文本，有 38 位短信作者总共发布了 100-499 条短信文本，有 11 位短信作者总共发布了 500 条以上的短信文本。

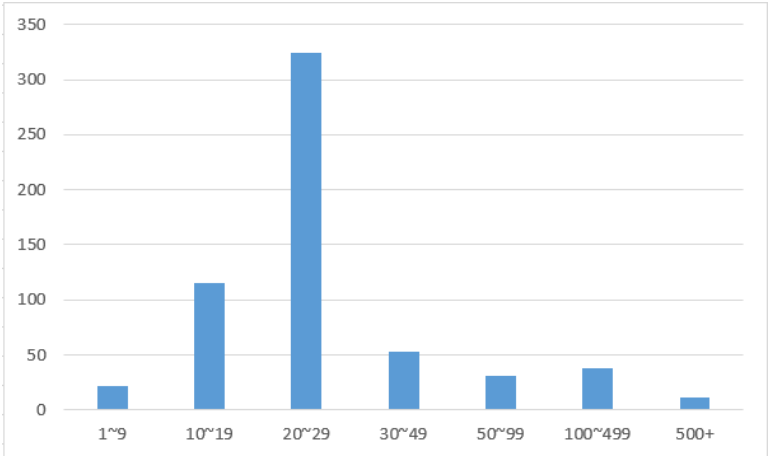


图 5.1 数据集中每个短信作者的短信文本数据量的分布情况

数据集中 86.5%的短信作者发布的短信文本在 50 条以下，属于相同作者的文本数据量较少，如果本文设计的用户认证模型能够较好的识别数据集中短信文本的作者，

即可预期该用户认证模型亦能在实际应用中取得较好的效果。

该数据集中有 61 位短信作者每天发送短信少于 1 条，有 30 位短信作者每天发送 1-2 条短信，有 194 位短信作者每天发送 2-5 条短信，有 152 位短信作者每天发送 5-10 条短信，有 131 位短信作者每天发送 10~50 条短信，有 12 位短信作者每天发送 50 条以上的短信，还有 21 位短信作者每天发送短信的频率未知。数据集中短信作者每天发送短信的频率的分布情况如图 5.2 所示。

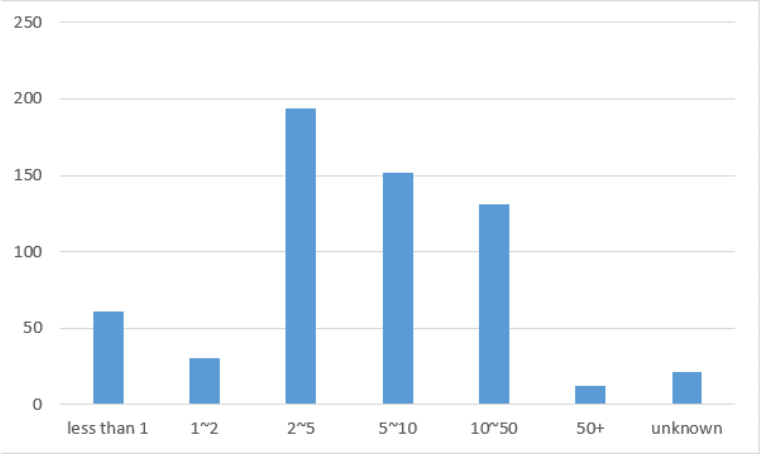


图 5.2 数据集中短信作者每天发送短信的频率的分布情况

该数据集中短信作者的年龄分布情况如图 5.3 所示，只有一位短信作者的年龄在 10-15 岁之间，有 72 位短信作者的年龄在 16-20 岁之间，有 344 位短信作者的年龄在 21-25 岁之间，有 110 位短信作者的年龄在 26-30 岁之间，有 22 位短信作者的年龄在 31-35 岁之间，有 5 位短信作者的年龄在 36-40 岁之间，有 2 位短信作者的年龄在 41-45 岁之间，有 9 位短信作者的年龄在 46-50 岁之间，也是仅有 1 位短信作者的年龄在 51-60 岁之间，有 35 位短信作者的年龄未知。

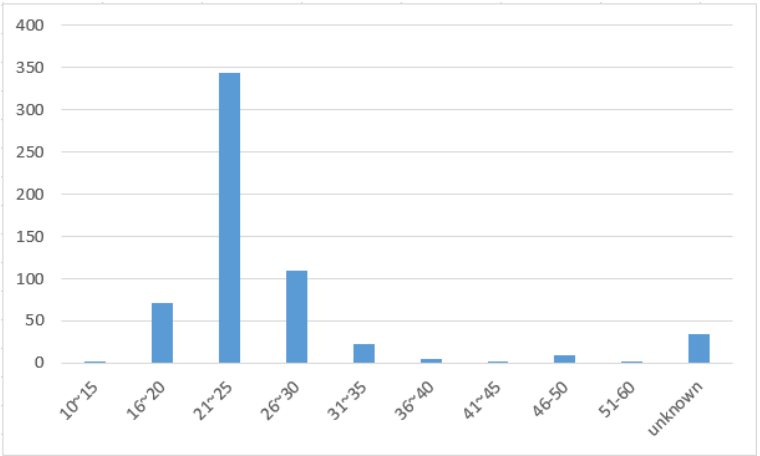


图 5.3 短信作者的年龄分布情况

数据集中文本作者的年龄分布情况与现今社交网络用户的年龄分布情况高度相似，

能够较好的评估用户认证模型在实际运行环境中的分类效果。

5.2.3 认证模型及判断阈值测试的测试数据

使用 5.2.2 节中介绍的公开数据集，按照以下算法制作训练集和数据集：

训练集和测试集制作算法

输入：NUS SMS Corpus 数据集原始数据

输出：训练集和测试集

1.按照 9:1 的比例将数据集中的数据随机划分为训练集和测试集

2.修改训练集，将训练集中每个用户对应的全部文本组合，构建用户对应的历史文本

3.修改测试集，修改随机划分后用户不在训练集中的数据，将此类数据的用户替换成训练集中的随机用户，标注为负样本

4.测试集中未修改的数据标注为正样本

5.统计正负样本数量，记为 num_pos 和 num_neg

6.当 num_pos>num_neg 时，平衡正负样本数量，待修改的样本数量为（num_pos-num_neg）/2 取整，记作 num_excess

7.在测试集中随机选择 num_excess 个正样本，将用户修改为训练集中出现的任意其他用户，并标记为负样本

由于原始数据集中仅包含文本作者、短信文本等信息，没有正负样本的区别，因此破坏一部分样本的文本作者和短信文本间的对应关系生成负样本，模拟用户被冒用的情况，其余文本作者和短信文本相对应的样本作为正样本。根据上述算法制作的训练集的基本情况如表 5.3 所示。

表 5.3 训练集的基本情况

数据总量	短文本作者数量
28318	590~594

在测试集中生成正负样本后，平衡两类样本的数量，消除样本分布对评估结果的影响，并基于上述算法开展 10 折交叉验证，全面地评估用户身份认证功能根据聊天信息区分正常用户和冒用者的能力。每轮交叉验证的测试集的基本情况如表 5.4 所示。

表 5.4 测试集的基本情况

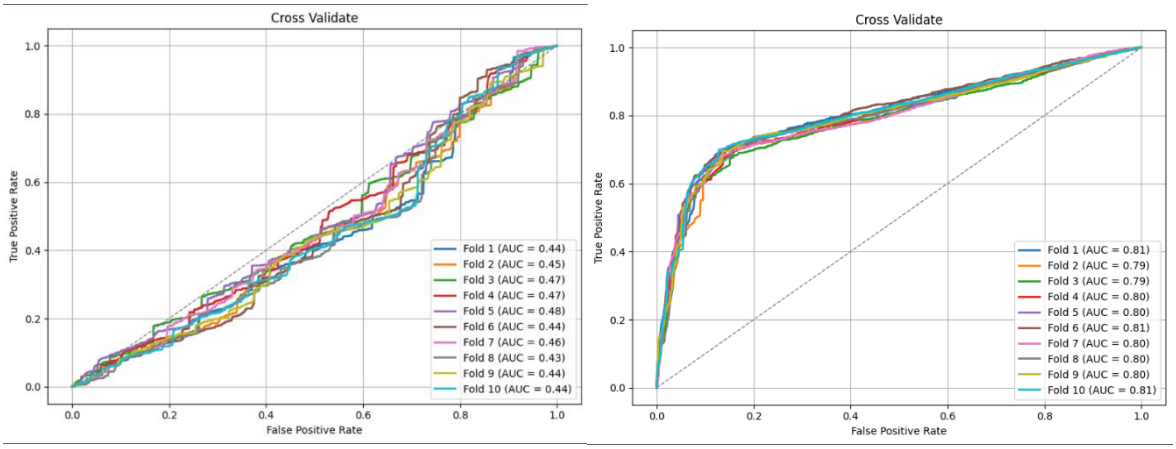
数据总量	短文本作者数量	正样本数量	负样本数量
3147	580~584	1573~1574	1573

5.3 功能性测试结果

根据 5.1.1 节功能性测试设计，本节按照认证模型及判断阈值的测试结果、认证结果处理功能的测试结果和文本特征更新功能的测试结果三个部分进行介绍。

5.3.1 认证模型及判断阈值的测试结果

使用 5.2.3 节制作的训练集和测试集，开展 10 折交叉验证，根据实验结果绘制的 ROC 曲线如图 5.4 所示。微调前认证模型的 AUC 值接近 0.5，不能根据聊天信息认证用户身份；微调后认证模型的平均 AUC 值为 0.8，能够根据聊天信息认证用户身份。



(a) 微调前模型测试结果的 ROC 曲线　　(b) 微调后模型测试结果的 ROC 曲线

图 5.4 微调前后认证模型测试结果的 ROC 曲线

模型微调前的判断阈值是 0.4982，模型微调后的判断阈值是 0.2063。微调前后用户身份认证模型的查全率、查准率、F1 值和 AUC 值如表 5.5 所示，微调后的用户身份认证模型根据样本标签分类的测试结果如表 5.6 所示。

表 5.5 微调前后用户身份认证模型的认证效果对比

模型	查准率 (Precision)	查全率 （Recall）	F1 值	AUC	测试样本数量
微调前的身份认证模型	0.57	0.52	0.42	0.45	31465
微调后的身份认证模型	0.78	0.77	0.77	0.80	31465

表 5.6 微调后的用户身份认证模型根据样本标签分类的测试结果

标签	查准率 （Precision）	查全率 （Recall）	F1 值	测试样本数量
0	0.73	0.87	0.79	15730

续表 5.6 微调后的用户身份认证模型根据样本标签分类的测试结果

标签	查准率（Precision）	查全率（Recall）	F1 值	测试样本数量
1	0.84	0.67	0.75	15735

根据表 5.5 和表 5.6 中的测试结果，本文设计的用户身份认证模型满足需求分析中查全率和查准率大于 0.7 的要求。同时，选择的判断阈值 0.2063 能够有效地区分正负样本，满足需求分析中判断阈值要能有效地区分用户本身和冒用者的要求。

5.3.2 认证结果处理功能的测试结果

在正常功能场景下，认证结果处理功能的测试用例和测试结果如表 5.7 所示。

表 5.7 认证结果处理功能的正常功能场景测试

用例编号	用例介绍	预期结果	测试结果
T-01	- 用户不在黑名单中 - 冒用者发送聊天信息	用户认证失败 被添加到黑名单中	通过
T-02	- 用户在黑名单中 - 冒用者发送聊天信息	前端删除用户 Token 强制退出用户 跳转至登陆页面	通过
T-03	- 用户在黑名单中 - 冒用者刷新聊天页面	前端删除用户 Token 强制退出用户 跳转至登陆页面	通过
T-04	- 用户在黑名单中 - 冒用者访问其他聊天页面	前端删除用户 Token 强制退出用户 跳转至登陆页面	通过
T-05	- 用户不在黑名单中 - 用户本人发送聊天信息	用户认证成功	通过

异常场景和边界条件下，认证结果处理功能的测试用例和测试结果如表 5.8 所示。

表 5.8 认证结果处理功能的异常场景和边界条件测试

用例编号	用例介绍	预期结果	测试结果
T-06	- 用户在黑名单中 - 用户本人发送聊天信息	前端删除用户 Token 强制退出用户 跳转至登陆页面	通过

根据表 5.7 和表 5.8 中的测试结果，本文设计的认证结果处理功能满足需求分析中的要求，当用户认证失败时触发二次验证。

5.3.3 文本特征更新功能的测试结果

在正常功能场景下，文本特征更新功能的测试用例和测试结果如表 5.9 所示。

表 5.9　用户文本特征更新功能的正常功能场景测试

用例编号	用例介绍	预期结果	测试结果
T-07	- 标志位 tag = 0 - 用户登陆时间与聊天信息发送时间之间的时间差为 0 分钟	修改聊天信息的标志位 tag = 1 保留聊天信息	通过
T-08	- 标志位 tag = 0 - 用户登陆时间与聊天信息发送时间之间的时间差为 2 分钟	修改聊天信息的标志位 tag = 1 保留聊天信息	通过
T-09	- 标志位 tag = 0 - 用户登陆时间与聊天信息发送时间之间的时间差为 4 分钟	删除聊天信息	通过
T-10	- 标志位 tag = 1 - 用户登陆时间与聊天信息发送时间之间的时间差为 4 分钟	保留聊天信息 不做任何处理	通过

异常场景和边界条件下，文本特征更新功能的测试用例和测试结果如表 5.10 所示。

表 5.10　用户文本特征更新功能的边界条件测试

用例编号	用例介绍	预期结果	测试结果
T-11	- 标志位 tag = 0 - 用户登陆时间与聊天信息发送时间之间的时间差为 3 分钟（等于阈值）	修改聊天信息的标志位 tag = 1 保留聊天信息	通过
T-12	- 标志位 tag = 0 - 用户登陆时间早于数据库中的聊天信息发送时间（时间差为负数）	删除聊天信息	通过
T-13	- 标志位 tag = 0 - 数据库中的聊天信息发送时间是未来时间	删除聊天信息	通过

根据表 5.9 和表 5.10 中的测试结果，本文设计的文本特征更新功能满足需求分析中的要求，能够根据用户登陆时间与聊天信息发送时间之间的时间差，保留正常用户的聊天信息，删除冒用者编写的聊天信息。

5.4 非功能性测试结果

根据 5.1.2 节非功能性测试设计，本节将按照兼容性测试结果、吞吐量测试结果和响应时间测试结果三个部分进行介绍。

5.4.1 兼容性测试结果

兼容性测试的用例和测试结果如表 5.11 所示。

表 5.11　持续身份认证机制的兼容性测试

用例编号	用例介绍	Chrome	Firefox	Edge
TC-01	- 聊天文本不存在乱码	通过	通过	通过
TC-02	- 聊天文本时间准确	通过	通过	通过
TC-03	- 删除用户 Token 功能正常	通过	通过	通过
TC-04	- 登出用户功能正常	通过	通过	通过
TC-05	- 无 Token 用户访问权限正常	通过	通过	通过

根据表 5.11 中的测试结果，本文设计的基于聊天信息的持续身份认证机制在常见的浏览器上能够正常运行。

5.4.2 吞吐量测试结果

吞吐量测试结果如表 5.13 所示。在最大并发量为 50、测试时长 3 分钟的场景下，本文设计的认证服务每秒能够处理 6.82 个请求，满足需求分析中在 5 分钟内处理至少 1000 个用户身份认证请求的要求。

表 5.13　持续身份认证机制的吞吐量测试

测试场景	并发数量	请求类型	请求增长速度	测试时长（分钟）	总请求数	吞吐量（RPS）	成功率
并发身份认证请求	10	POST	2	1	300	5	100%
	50		10	3	1228	6.82	100%
	100		25	10	3992	6.65	97.1%

在最大并发量为 100、测试时长为 10 分钟的场景下，本文设计的认证服务每秒能

够处理 6.65 个请求，但受限于实验平台的硬件能力，请求成功率为 97.1%。

5.4.3 响应时间测试结果

响应时间测试结果如表 5.14 所示。在最大并发量为 50、测试时长 3 分钟的场景下，本文设计的认证服务的平均响应时间为 5633.85 毫秒（5.63 秒），最大响应时间为 27760 毫秒（27.76 秒）。

在最大并发量为 100、测试时长为 10 分钟的场景下，本文设计的认证服务的平均响应时间为 13320.11 毫秒（13.32 秒），最大响应时间为 126068 毫秒（126 秒）。满足需求分析中在同时处理 100 个并发请求的前提下，平均响应时间不超过 15s，最大响应时间不超过 3 分钟的要求。

表 5.14　持续身份认证机制的响应时间测试

测试场景	并发数量	请求类型	请求增长速度	测试时长（分钟）	平均响应时间（毫秒）	最大响应时间（毫秒）	最小响应时间（毫秒）
身份认证	10	POST	2	1	444.37	3230	36
	50		10	3	5633.85	27760	581
	100		25	10	13320.11	126068	738

5.5 本章小结

首先，本章通过实验和测试设计介绍了功能性测试和非功能性测试的开展方案，然后本章补充介绍了测试环境和测试数据。最后，本章介绍了功能性测试和非功能性测试的结果，相对完整地测试了本文设计的持续认证机制。

第六章　结论与展望

6.1 结论

本文从用户在线聊天的过程中缺乏持续的身份验证手段保障用户的账号安全这一问题切入，对比了不同的持续认证方案的优点和不足，最终选择设计并实现一种基于聊天信息的持续身份认证机制解决上述问题。

通过比较不同模型组合基于聊天信息进行用户身份认证的能力，本文最终选择使用 BERT-BiGRU-Attention(softmax)结构的模型实现用户身份认证功能。然后，本文使用 Vue2 和 Spring Boot 框架实现了一个简单的聊天服务的前后端，使用 Fast API 框架实现了认证机制的后端，完成了基于聊天信息的持续身份认证机制的设计和实现。

最后，在持续身份认证机制的测试中，本文从功能性测试和非功能性测试两个方面评价并完善了本文设计的基于聊天信息的持续身份认证机制。微调后的用户身份认证模型在公开数据集上的测试结果是，查准率 0.78，查全率 0.77，F1 值 0.77，根据测试结果绘制的 ROC 曲线的平均 AUC 值为 0.8，持续认证机制能够基于聊天信息认证用户的身份。

综上所述，本文完成了一种基于聊天信息的持续身份认证机制的设计和实现。

6.2 展望

本文使用的公开数据集中记录有部分短信文本的详细发送时间和短信作者间的聊天关系，但本文设计的持续身份认证机制没有充分利用聊天内容的上下关联信息、聊天信息的时序等关键特征训练模型和开展实验。

之后开展基于聊天信息的持续身份认证机制的相关研究时，应充分利用上述关键特征训练模型，选择包含聊天信息发送者、聊天信息接收者、聊天文本和聊天信息发送时间四个属性的数据集，并且还要确保数据集中的聊天信息发送者和聊天信息接收者存在一定数量的重叠。

参考文献

[1] 腾讯控股有限公司. 腾讯控股有限公司2024年报[R/OL]. 香港: 腾讯控股有限公司, 2025[2025-04-08]. 投资者 - Tencent 腾讯

[2] 中国互联网络信息中心. 第 54 次中国互联网络发展状况统计报告[R/OL]. 北京: 中国互联网络信息中心, 2024[2024-08-29]. 第 54 次《中国互联网络发展状况统计报告》--互联网发展研究

[3] KAUR R, RAVNEET, SINGH P, et al. TB-CoAuth: Text based continuous authentication for detecting compromised accounts in social networks[J]. *Appl Soft Comput*, 2020, 97: 106770.

[4] BAIG A F, ESKELAND S. Security, Privacy, and Usability in Continuous Authentication: A Survey[J]. *Sensors*, 2021, 21(17): 5967. https://doi.org/10.3390/s21175967.

[5] CROUSE D, HAN H, CHANDRA D, et al. Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data[C]// *Proceedings of the 2015 International Conference on Biometrics (ICB)*. Phuket, Thailand: IEEE, 2015: 135–142.

[6] FENG H, FAWAZ K, SHIN K G, et al. Continuous authentication for voice assistants[C]// *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. Snowbird, UT, USA: ACM, 2017: 343–355.

[7] GOMI H, YAMAGUCHI S, TSUBOUCHI K, et al. Continuous authentication system using online activities[C]// Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). New York, NY, USA: IEEE, 2018: 522–532.

[8] DERAWI M O, NICKEL C, BOURS P, et al. Unobtrusive user-authentication on mobile phones using biometric gait recognition[C]// *Proceedings of the IEEE 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. Darmstadt, Germany: IEEE, 2010: 306–311.

[9] BROCARDO M L, TRAORE I, WOUUNGANG I. Authorship verification of e-mail and tweet messages applied for continuous authentication[J]. *J Comput Syst Sci*, 2015, 81: 1429–1440.

[10] GASCON H, UELLENBECK S, WOLF C, et al. Continuous authentication on mobile devices by analysis of typing motion behavior[C]// *Sicherheit 2014–Sicherheit, Schutz und Zuverlässigkeit (Lecture Notes in Informatics (LNI))*. Vienna, Austria: Gesellschaft für Informatik (GI), 2014: 1–12.

[11] MENDENHALLTC．The characteristic curves of composition[J]．Science，1887，9(124)：237—249．

[12] 张洋，江铭虎．作者识别研究综述[J/OL]. 自动化学报:1-26[2021-04-01].https://doi.org/10.16383/j.aas.c200654.

[13] Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A framework for authorship identification of online messages: Writing-style features and classification techniques[J]. *J. Assoc. Inf. Sci. Technol., 57*, 378-393.

[14] 吕英杰，范静，刘景方．基于文体学的中文UGC作者身份识别研究[J]. 现代图书情报技术. 2013,(09):48-53.

[15] 祁瑞华，杨德礼，郭旭，等．基于多层面文体特征的博客作者身份识别研究[J]．情报学报，2015，34(6)：628—634．

[16] YANG M，ZHU D，TANGY，et a1．Authorship attribution with topic drift model[C] //Proceedings of the Thirty First AAAI Conference on Artificial Intelligence，2017：5015—5016．

[17] 徐晓霖，蔡满春，芦天亮．基于深度学习的中文微博作者身份识别研究[J]．计算机应用研究，2020，37 (1)：16—18，25．

[18] ZHANGR，HUZ，GUO H，et a1．Syntax encoding with application in authorship attribution[C] //Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing，2018：2742—2753．

[19] 冯勇, 屈渤浩, 徐红艳, 等. 融合 TF-IDF 和 LDA 的中文 FastText 短文本分类方法[J]. 应用科学学报, 2019, 37(3)：378-388. DOI:10.3969/j.issn.0255-8297.2019.03.008.

[20] 张翼翔, 芦天亮, 李默. 基于 BERT-BiGRU-ATT 的社交媒体用户身份识别研究[J]. 中国人民公安大学学报（自然科学版）,2021,27(1):70-75

[21] CHEN T, KAN M Y. Creating a live, public short message service corpus: The NUS SMS Corpus[J]. *Lang Resour Eval*, 2013, 47(2): 299–355.

[22] 李孟林, 罗文华, 李绍鸣. 基于神经网络中文短文本作者识别研究[J].中国人民公安大学学报：自然科学版, 2020, 26(2):7.

[23] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]// *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, MN, USA: Association for Computational Linguistics, 2019: 4171–4186.

[24] MNIH V, HEESS N, GRAVES A, et al. Recurrent models of visual attention[C]// *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*. Montreal, Canada: MIT Press, 2014: 2204–2212.

[25] Liu W, Wen Y, Yu Z ,et al. SphereFace: Deep Hypersphere Embedding for Face Recognition[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).IEEE, 2017.DOI:10.1109/CVPR.2017.713.

致谢

值此论文完成之际，向曾经在学习生活中帮助过我的各位致以真挚的感谢。

首先，感谢孙惠平老师提供的选题和指导。在完成本文的过程中，我初步接触了持续认证机制与自然语言处理这两个研究领域，开展了模型设计、模型实现和参数微调的实践。这段经历不仅加深了我对人工智能技术的理解，也为我未来的学习探索和职业规划指明了方向。

其次，感谢我的父母和好友在生活中给予我的关怀和帮助，目前我取得的这些成果都离不开各位的支持。

最后，也感谢北京大学软件与微电子学院的软件工程第二学士学位项目，这两年软件工程专业知识的学习使我受益良多。