目　录

摘　　要

随着互联网技术的不断发展，供应链管理逐渐成为电商平台的核心支撑点。供应链旨将商品流通到各个节点所构成的链式网络结构，其中包含供应商、经销商、消费者等多个实体角色，同时包含数据流、资金流、物流等多种信息交互。因此通过对供应链数据的收集整理构建完全信任的数据共享空间是促进电商供应链多方协作和提高供应链效益的关键。

近年来，电商平台已经逐步成为人们生活中不可缺少的一部分，因此出现许多消费者反应商品信息与店铺销售信息不符的情况，导致消费者对线上购物的信任度逐渐降低。同时电商平台无法对供货商铺进行更有效的监管。因此区块链技术去中心化、可溯源性、不可篡改性的应用能有效解决电商供应链行业中的信任问题，但尽管将数据共享在同一环境中协同使用，也衍生出新的痛点和不足：一方面，区块链的冗余存储特性，容易使区块链网络造成压力过载的情况；另一方面，由于大量数据的叠加存储，无法为供应商提供准确的商户运营状况，也无法对需求量数据进行精准预测分析，降低了供应链管理的效率。

综上所述，解决区块链网络存储压力过载的问题和提高供应链需求量预测精准度的问题，使构建基于区块链的电商供应链交易平台是极具现实意义的。因此，本文的主要研究内容如下：

（1）提出面向电商供应链的区块链存储方案，在电商供应链交易过程中会产生大量的交易数据，由于电商数据量较大，导致数据存入区块链网络中造成数据存储过载的问题。基于 IPFS 技术将供应链数据先进行分类压缩再进行异构处理，以此缓解数据上链存储的压力。为进一步降低区块链的存储压力，提出链下同步机制，将电商供应链的详细数据存储同步存储至链下数据库，利用 SM2 数字签名机制确保链上链下数据存储的一致性及安全性。经测试证明这种链上链下存储方案的设计能够有效缓解区块链网络中的存储压力。

（2）提出基于区块链的电商供应链需求量预测模型，借助以太坊的数据特性实时收集区块链中的交易数据，并分布进行数据预处理，再根据经特征工程优化的 LSTM 算法构建需求量预测模型，结合智能合约对账户角色进行监控，不仅提高了预测模型的精准度还确保了数据信息的安全性。

（3）构建电商供应链交易平台，利用以太坊搭建区块链网络，借助 IPFS 对大文件数据进行分布式存储，再结合 GO 和 Python 实现系统的业务功能及模型预测，后端采用 BeeGo 框架进行服务端搭建，前端采用 LayUI 进行页面交互，最后进行功能测试，保障系统性能的有效性。

关键词　电商供应链；链上链下存储方案；优化 LSTM 算法预测；以太坊

ABSTRACT

With the continuous development of Internet technology, supply chain management has gradually become the core support point of e-commerce platform. The supply chain refers to a chain network structure formed by the circulation of goods to various nodes, which includes multiple physical roles such as suppliers, distributors, and consumers, as well as various information interactions such as data flow, capital flow, and logistics. Therefore, building a fully trusted data sharing space through the collection and organization of supply chain data is the key to promoting multi-party collaboration in e-commerce supply chains and improving supply chain efficiency.

In recent years, e-commerce platforms have gradually become an indispensable part of people's lives. As a result, many consumers have reported that product information does not match store sales information, leading to a gradual decrease in consumer trust in online shopping. At the same time, e-commerce platforms are unable to provide more effective supervision of suppliers. Therefore, the decentralized, traceable, and tamper proof application of blockchain technology can effectively solve the trust problem in the e-commerce supply chain industry. However, although data sharing and collaborative use in the same environment also give rise to new pain points and shortcomings: on the one hand, the redundant storage characteristics of blockchain can easily cause pressure overload in the blockchain network; On the other hand, due to the stacking and storage of a large amount of data, it is impossible to provide accurate merchant operation status for suppliers, and it is also impossible to accurately predict and analyze demand data, which reduces the efficiency of supply chain management.

In summary, it is of great practical significance to build a blockchain based e-commerce supply chain trading platform by solving the problem of storage overload in blockchain networks and improving the accuracy of supply chain demand prediction. Therefore, the main research content of this article is as follows:

(1) Propose a blockchain storage solution for the e-commerce supply chain, which generates a large amount of transaction data during the e-commerce supply chain transaction process. Due to the large amount of e-commerce data, it leads to the problem

of data storage overload when stored in the blockchain network. Based on IPFS technology, supply chain data is first classified and compressed before heterogeneous processing, in order to alleviate the pressure of data on chain storage. To further reduce the storage pressure of blockchain, an off chain synchronization mechanism is proposed, which synchronously stores detailed data of e-commerce supply chain to the off chain database. The SM2 digital signature mechanism is used to ensure the consistency and security of on chain and off chain data storage. Tests have shown that the design of this on chain and off chain storage solution can effectively alleviate storage pressure in blockchain networks.

(2) Propose a blockchain based e-commerce supply chain demand prediction model, utilizing the data characteristics of Ethereum to collect real-time transaction data in the blockchain and distribute it for data preprocessing. Then, based on the LSTM algorithm optimized by feature engineering, a demand prediction model is constructed, and combined with smart contracts to monitor account roles. This not only improves the accuracy of the prediction model but also ensures the security of data information.

(3) Build an e-commerce supply chain trading platform, use Ethereum to build a blockchain network, use IPFS for distributed storage of large file data, and combine GO and Python to achieve system business functions and model prediction. The backend uses the BeeGo framework for server building, and the frontend uses LayUI for page interaction. Finally, functional testing is conducted to ensure the effectiveness of system performance.

Key words: E-commerce Supply Chain; On Chain and Off Chain Storage Solutions; Optimizing LSTM Algorithm Prediction; Ethereum

第一章 绪论

1.1 研究背景及意义

随着互联网技术逐步在全球普及和经济全球化发展趋势的转变[1]，商品贸易的形式也逐渐变的多元化，由传统的线下企业、个人交易向电商网络转变，而交易的商品也逐步个性化、碎片化，不再以大众普遍化为主。

近年来，电子商务的市场规模逐渐扩大，根据 2023 年星图数据发布的《2023 年电商发展报告》显示，我国 2022 年电子商务的市场规模已达到 41.8 万亿元，同比增长 16.4%，占 GDP 比重达 36.8%。我国电商行业的蓬勃发展离不开供应链的支撑，供应链是指在商品流动期间所涉及的核心企业、商户和消费者之间共同构建的网状结构。电商供应链是一个典型的分布式应用场景，其中不仅涉及供应商、分销商和消费者等多个角色实体，同时还包括物流、资金流和信息流等多种数据类型。由此可见，在不全完全信任的多种角色之间整合繁杂的数据流并建立共享环境，是促进供应链整体效益的关键。同时，供应链共享环境的构建，便于提高商品数据流的准确性，有助于提高供应链数据需求量预测的精准度。

区块链技术是通过多种技术融合形成的，这使区块链具有去中心化、不可篡改、可追溯等特性，这些特点使得其在供应链信息追溯方面有了不可替代的优势[2]。区块链技术可以在完全不信任的交易节点中，通过加密算法、共识机制等方式完成数据的交易并将数据存储于可信环境中，这使得区块链的分布式存储比集中式存储更可靠[3]。随着区块链技术的不断发展，大众普遍认为区块链已经完全可以改变现有的信任问题[4]。因此，近年来关于区块链供应链的研究也逐渐增多，有关区块链供应链平台的数据安全性、透明性、准确性等相关问题也成为了当下学者研究的热点。

然而，利用区块链的理论知识及博弈模型能在一定程度上实现零售商和供应商之间数据信息共享[5]，但对区块链上的数据存储压力及数据隐私性并未涉及。而针对供应链中需求量预测的准确性，采用博弈论模型分析当下消费者的低碳意识行为与产品需求量、产品利润、制造商最优利润之间的关系，使供应链中不同主体可以决定最优成本分担比例从而实现帕累托改进并提高供应链效率[6]，这虽然改善了供应链的

预测分析方式，但未能提高需求量预测的精准度。

综上所述，探索如何利用区块链技术构建数据信息共享环境，在缓解链上存储压力的同时提高供应链数据使用多元化的电商供应链交易平台是非常具有现实意义的课题。基于上述现状，本文首先针对区块链的存储结构结合 IPFS 技术提出面向电商供应链的链上分类异构存储方案，并结合加密算法提出链下数据认证同步机制。其次针对供应链数据使用单一性的问题，结合数据分析技术将数据预测模型引入电商供应链平台，提出了基于特征工程的 LSTM 预测优化方案并结合智能合约，提高供应链交易平台商品信息的准确性及预测数据的精准度。最后综合上述两种方案构建基于区块链的电商供应链交易平台，在确保数据准确性的同时降低区块链的链上存储压力，提高商品需求量预测的精准度。最大程度实现数据的公开、透明，减轻消费者对电商平台的质疑，规范企业商家的供应运营流程，促进上下游企业之间的协同合作，提升企业对市场的决策力度及商品的市场定位。

1.2 国内外研究现状

1.2.1 区块链技术研究现状

目前，区块链技术的研究在国内外的研究现状中依旧处于热点话题，国外的研究主体依旧是以比特币和以太坊为主，比特币是一种运用密码学原理构建去中心化的数字货币，其核心是利用了区块链的去中心化特性在公共账簿上实现点对点的交易[7]；而以太坊则是在比特币的基础之上引入智能合约技术，将其转化为可编程的形式[8]，为区块链个性化设置提供了条件。区块链技术的不断提升，为我们构建完全可信的网络环境提供了更高的效率[9]。人们对区块链的完善不仅是基础形态的调整，还有局部技术的优化，例如对共识机制的研究，Manpreet Kaur 等[10]对工作量证明、权益证明、活动证明等多种主流共识机制展开了详细的研究和性能分析，提出了在不同场景下选用各类共识机制的参考指标。Cristain Lepore 等[11]利用权益证明（PoS）共识算法对工作量证明（PoW）共识算法进行优化，一定程度上解决了吞吐量，但仍需要进一步改善和解决节点集中化的现象。Lee DR 等[12]提出基于分片的 PoS 区块链协议，克服了现有分片共识协议的局限性。Alhejazi M M 等[13]利用分散区块链来加权多个共识算法，以提高数据的安全性。在区块链存储方面，Agarwal V 等[14]提出使用侧链和线下数据存储的概念来缓解区块链延展性的问题，利用物联网系统的安全架构结合智能合约缓解庞大的数据存储压力。

对于国内的研究而言，虽然区块链技术在国内的认知较晚，但发展速度还是很快，张长贵等[15]针对图型和分区型两种新型区块链系统进行了详细的分析和研究发现，前者实时性很强，后者伸缩性较好。王谨东等[16]利用 Raft 算法对拜占庭共识算法进行了改进，在一定程度上提高了共识效率和吞吐量。梁昊等[17]利用联盟链构重构了区块链的框架，使数据存储方式更加灵活。李莉等[18]利用融合区块链和代理重加密技术，保障了数据流通间的安全性。周家栋[19]设计了一种融合 LZ78 和哈夫曼编码的压缩组合方法（LZ-Hf），对溯源数据进行有效压缩，缓解了农产品溯源信息中数据的冗余性。

1.2.2 基于区块链的供应链系统研究现状

随着区块链技术的不断发展和创新，该项技术已经被广泛应用到供应链、金融、医疗等多个领域。2016 年，我国《“十三五”国家信息规划》提出将区块链作为重点前沿技术，需要加强相关的技术创新和应用，才能抢占新一代信息技术的主导权[20][21]。2019 年，中央政治局第十八次集体学习时强调，应把区块链技术作为自主创新的核心突破口，促进区块链技术和产业的发展[22]。国家邮政局、工业和信息化部在 2020 年 4 月发布的《关于促进快递业与制造业深度融合发展的意见》中提高要深化区块链技术与制造业供应链的融合，促进制造企业和快递企业的发展[23]。国家先后出台的一系列政策提高了区块链在供应链管理中的应用价值。

区块链技术的不可篡改性、透明性、可溯源性为数据的交易和维护提供了更完善和保密的机制，同时也解决了供应链管理中的许多问题，提高了供应链的运行效率。例如，白燕飞等[24]利用区块链技术重构了供应链金融平台，通过优化企业的互信机制，提高了融资的安全性和效率。Jangirala 等[25]对 5G 移动边缘计算背景下的供应链进行分析，提出基于区块链和 RFID 的认证协议，保障了供应链间数据传输的安全性，节约了数据通讯间的计算成本。庄楚鑫[26]借助区块链技术设计出中心化的医疗药品供应链协同管理系统，解决了供应链数据易篡改、不共享等问题。徐俊等[27]将物联网终端设备与区块链相融合，实现了冷链流程中的数据采集、上传功能自动化。Ivanov D 等[28]通过将供应链融合区块链进一步研究发现，区块链技术提高了供应链数据的可用性并降低了数据交互的成本。许蕴韬等[29]基于 DPoS 共识机制提出选举供应链，促进了供应链的协同作用。周欣[30]通过分析区块链在跨境供应链和海关合规领域的国内外优势，提出官企合作、数据信息协同的应用创新建议。戴艳、朱方[31]提出利

用区块链技术重新构建跨境电商的信任体系，解决了跨境电商物流跟踪、质量控制、产品溯源等多个信任痛点，促进了跨境电商的进一步发展。

1.2.3 需求量预测的研究现状

近年来，国内外关于需求量预测的方法主要分为时间序列算法和人工智能算法两大类。但在实际的研究中，研究人员还是会根据实际情况选用对应的模型进行优化，以提升预测的准确性和效率。例如，Torbat S 等[32]基于模糊自回归集成移动模型提出了一种混合模型，利用概率分类器识别线性模式缩小了传统自回归集成移动模型的间隔。HASHIM F A 等[33]对阿基米德算法进行优化，提高了模型的收敛速度。HONG J-K[34]利用 LSTM 预测模型对温度敏感产品的销量进行预测，比传统的预测方法更容易捕捉数据的特征值。PUNIA S 等[35]将 LSTM 与 RF 融合设计出多渠道零售预测模型，一定程度上提高了预测结果的准确度。苗晓峰等[36]提出了一种 LSTM 与时空结合的方法对不同空间类型的站点进行车辆需求量预测，相较于传统基于历史出行数据、时间和天气数据的预测模型精准度更高、性能更好。王辉、李昌刚[37]通过研究 Stacking 模型的特性，并在此基础上融合多个机器学习模型构建出新的需求量预测模型，提高了在供应链中的应用价值。包吉祥等[38]将消费者行为和历史数据引入 LSTM 算法中对快消产品的需求量进行预测，虽然这种方式考虑到了电商产品的滞后性且提高了预测的精准度，但由于数据对象选取的局限性，还需做进一步的研究和改进。王慧萍[39]以 WL 公司为例，提出利用区块链技术优化产品的营销策略，为企业实施精准营销提供保障。施亚东[40]利用区块链技术设计金融产品推送模型，提高了产品的推送速度和精度，具有更高的实用价值。

1.3 论文内容及章节安排

本课题主要研究在电商供应链领域区块链技术和大数据分析技术相结合的应用，旨在运用区块链技术确保电商供应链信息的透明性和真实性，利用密码学技术对供应链交易数据隐私性的掌控以及借助数据分析技术对供应链需求量的精准预测。论文的章节安排如下：

第一章 绪论。介绍研究电商供应链的重要意义，通过对区块链技术在供应链领域的研究背景分析及国内外研究现状分析，强调了区块链应用于电商供应链平台和高效数据分析的必要性。

第二章 相关理论与技术。对基于区块链的电商供应链交易平台所涉及的相关技

术进行研究。首先是对区块链技术本身进行介绍，重点阐述了区块链的存储机制及智能合约的应用。接着介绍了以太坊区块链网络背景下供应链数据存储和加密技术。最后介绍了供应链需求量预测的相关技术。

第三章 面向电商供应链的区块链数据存储方案。对电商供应链数据在区块链中的存储方式和数据隐私进行分析，在此基础上提出了链上数据异构存储方案和链下数据同步机制，通过对数据进行分类和加密，减缓区块链网络数据的存储压力并加强数据防篡改的能力。

第四章 基于区块链的电商供应链需求量预测模型。通过以太坊区块链网络实现数据的透明、共享，对供应链交易数据的实时收集和预处理，确保数据的有效性和可用性，利用特征工程对 LSTM 算法进行优化，并嵌入进区块链中实现分布式数据整理预测，再融合智能合约技术对区块链网络中的账户进行权限管控，确保在提高模型预测精准度的同时保障数据的安全性。

第五章 基于区块链的电商供应链平台的实现与测试。首先对电商供应链交易平台的总体架构进行设计，并在此基础上进行数据库的建设。接着再将第三章和第四章中的存储机制及预测模型融合进系统，配置相关的协议和文件，进一步实现系统的各个模块功能及展示。最后完成对系统的测试。

第六章 总结与展望。总结本课题所完成的各项内容，并提出未来可以将此研究作为基础进行研究的内容和方向。

第二章 相关理论与技术

2.1 区块链相关技术

2.1.1 区块链基本概念

区块链的概念诞生于比特币，最初是由中本聪在 2008 年发表的比特币白皮书中提出的[41]。它是一种基于密码学算法的点对点分布式账本技术，且同时包含了分布式存储、共识机制、加密算法等多个计算机技术的新型应用模式[42]。

区块链的基本要素包含交易、区块和链三个。交易用于记录每次操作完成后账本状态所发生的变化；区块主要是记录在一段时间内所发生的交易和状态结果，这是对当前账本状态的一次共识；而链是有由多个区块按时间顺序串联而成，用于表示整个状态的日志记录。区块链是一个收集了所有历史交易记录的分布式公开数据账本[43],具有信息透明、去中心化、可追溯、不可篡改等特点[44]。

区块链是以区块为基本单元的链式存储结构，如图 2.1 所示每条链都是由多个区块构成，每个区块都是由区块头和区块体两部分组成[45]。区块头用于存放前一个区块的哈希值、Merkle 根、时间戳等数据，区块体用于存储从上一个区块的时间戳到本区块的时间戳的这段时间内所发生的交易信息。

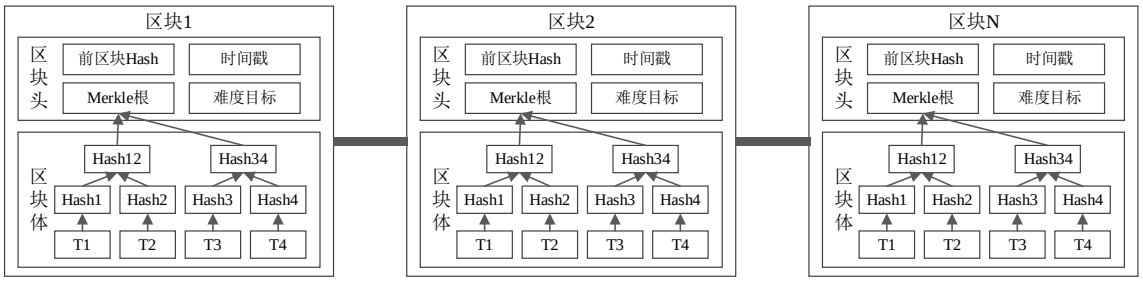

图 2.1 区块链结构图

区块链的基础体系架构有六层，如图 2.2 所示从下到上依次是数据层、网络层、共识层、激励层、合约层和应用层。数据层封装了底层的数据区块以及基本的数据加密算法，用于确保数据存储和交易的安全。网络层具有分布式网络结构、数据验证和传播机制等，确保数据节点间的有效通信。共识层中含有多个共识算法，包括

PoW、PoS、DPoS 等多个共识机制，帮助实现交易数据的打包和上链。激励层主要包括激励的发行和分配机制，为区块链中参与安全验证的节点提供奖励。合约层主要是封装了各类算法、代码脚本和智能合约，为账本提供可编程的属性。应用层包含了区块链的各类应用场景和案例。

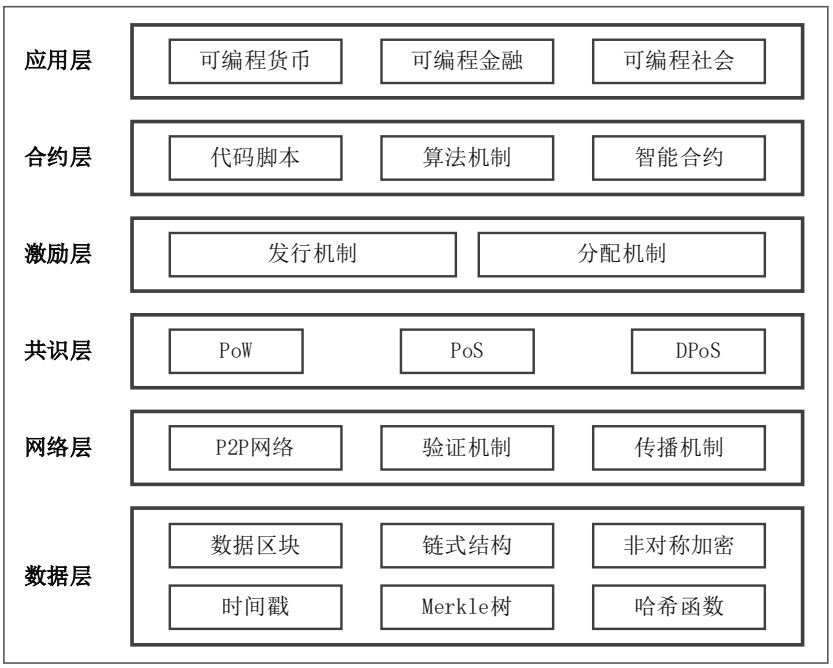

图 2.2 区块链基础体系结构图

2.1.2 以太坊

以太坊最初是 Vitalik Buterin 受到比特币的启发在《以太坊：一个下一代智能合约和去中心化应用平台》中提出的[46]，它是首个支持智能合约的区块链平台[47],也是众多公有链中的一种。以太坊的本质是基于交易的状态机，利用区块链来同步和存储系统状态以及利用以太币加密货币。以太坊中的数据交易是由以太坊内部节点组成的以太坊主网,通过主网协议在网络中实现自主交易,并借助以太坊虚拟机（EVM）完成数据间的信息传输。

以太坊网络中的账户分为两类：用户账户和合约账户。用户账户是指在加入区块链网络时为节点创建的账户，这个账户由一对公私钥所控制。合约账户是存储在区块链中的一个地址代码，便于合约交易中账户的指定。这两类账户的区别在于，用户账户中代码和存储均为空，它只是各个节点在网络中的交易凭证。而合约账户中包含了存储和代码，只允许被用户账户进行调用，不能主动向其他账户发送交易。

在以太坊中还引入了一个重要元素：Gas，用户账户无论在网络中进行转账交易

还是合约调用都会产生一定的 Gas 消耗，这是为了对用户请求的工作量进行资源限制，同时这笔消费也会随着交易的成功而奖励给链上的矿工。Gas 是以太坊中的燃料，用于保证以太坊生态的正常运作。当 Gas 全部用完，但交易未完成时，网络交易中会给出用户警示，并不在退还已消耗的 Gas，这种方式能够避免一些大量算力的消耗和无效交易的计算。

2.1.3 智能合约

智能合约的概念最初是由 Nick Szabo 在 1994 年提出的[48]，提出这一概念的目的是为了给传统合约提供更安全的方案，同时减少与合约相关的其他交易成本。但由于当时缺少可信的网络环境，这一个概念并未得到真正的实施，而区块链的出现让智能合约的实行成为了可能[49]，在以太坊的区块链网络环境中，将以太坊的自身技术与这一概念融合，利用 Solidity 完成智能合约中相关交易规则的构建，并通过虚拟机将合约上链运作，不需要对合约进行人为的控制，通过对特定触发点的设置，合约就可以自动执行。这种方式大大减少传统合约的复杂性，保障了合约的安全有效。

完整的智能合约主要包括制定、部署、交易和作废这四个部分，每个步骤都紧密相关，以确保合约的顺利进行。智能合约的运行机制如图 2.3 所示，首先用户根据自身的具体业务需求编写智能合约的相关内容并生成.sol 文件，然后通过以太坊的虚拟机进行编译，使.sol 文件变成可操作的字节码文件，最后将其部署到 EVM 中，使交易信息能够按规则在区块链网络间相互传输。

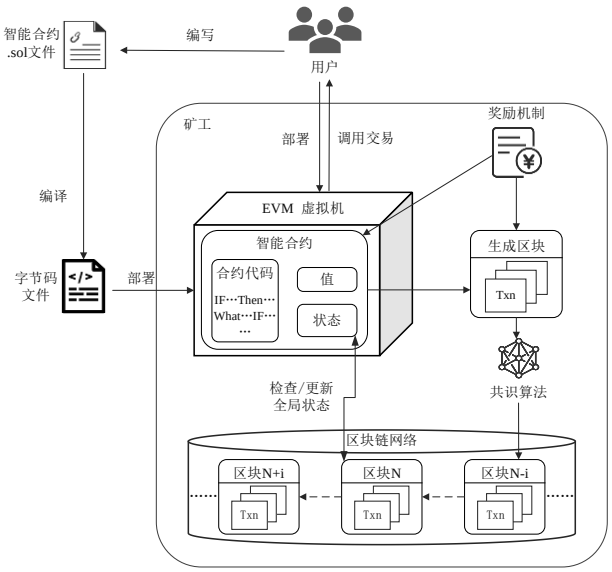

图 2.3 智能合约运行机制图

2.2 电商供应链交易存储加密相关技术

2.2.1 以太坊区块存储结构

在以太坊中，数据的存储结构和验证是非常关键的环节，数据的存储大致分为三个部分，分别是状态数据、区块链和底层数据。其中，底层数据以键值对的形式存放以太坊中，目前常用的数据库是 LevelDB，而所有与交易、操作相关的数据都存储在链上；状态数据是利用 StateDB 来管理账户的，且每一个账户都是一个 StateObject；以太坊中的区块存储结构是其核心之一，所有的交易和机构都存储于一个个区块中，主要包括区块头和区块体两部分。区块体存储了区块的元信息，这些信息用于对区块内容进行标识、校验和说明。主要包含了时间戳、上一区块的哈希值、消耗的 Gas、状态树的根哈希值、交易树的根哈希值和收据树的根哈希值等数据信息。状态树是 Merkle Patricia Tree，用于维护整个系统中账户的状态，交易树和收据数是 Merkle 树，用于记录每次的交易信息和对应的收据。如图 2.4 所示，在以太坊的区块存储结构中，每个区块头都会建立状态树、交易树和收据树的根值构成映射关系，以确保所有节点交易存储状态的一致性。

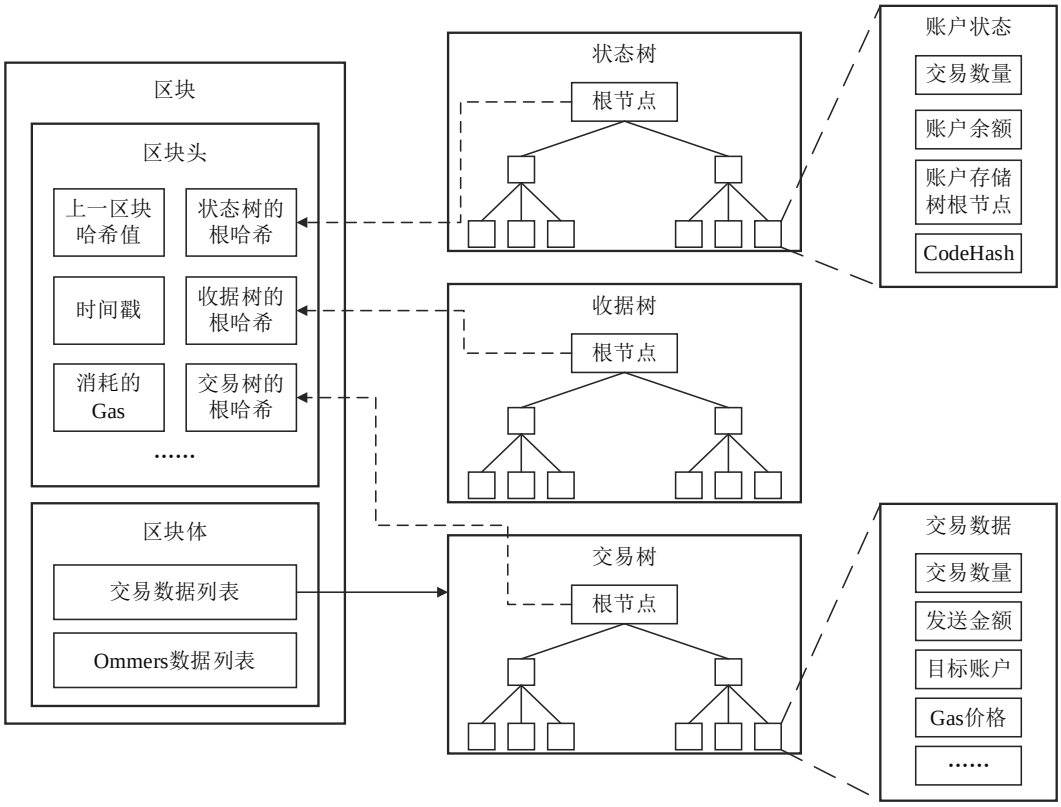

图 2.4 以太坊的区块存储结构图

在每次交易后，以太坊都会先将交易存储至交易池中生成新的区块，在区块通过挖矿生成并达成全网共识后，再将交易池中的交易打包成区块并全网广播，而所有通过节点验证的区块会连接至以太坊区块链的尾部，使每笔交易记录都是可追溯的。因此，本文利用以太坊的存储结构搭建电商供应链交易平台，商品在供应链中的每笔交易的有迹可循，便于账户间的数据访问和确认。

2.2.2 星际文件系统基本概念

星际文件系统（IPFS）是一种点对点的分布式文件系统，它旨在创建更快、更安全、全球化的存储和共享网络。IPFS 是基于分布式哈希表 DHT、Git 模型、默克尔对象关联、点对点技术等多种技术，将文件加密后进行碎片化处理，并将碎片分散存储在存储器中，实现内容寻址和分布式文件系统。IPFS 中的每一个文件都附有唯一的哈希值，且与文件的具体位置无关，用户只需获取文件对应的 hash 值就能快速找到文件，提高了文件的传输效率和安全性[50]，同时有效地消除了重复性文件，降低了资源消耗的成本。此外，IPFS 还具有自我修复的特性。当某些节点出现问题或离线时，其他节点可以自动接管这些节点的工作，确保系统的稳定运行，这使得 IPFS 具有很高的可靠性和稳定性。若将区块链技术与 IPFS 相结合，在一定程度上能缓解链上的存储压力。

2.2.3 SM2 加密算法

SM2 加密算法是基于椭圆曲线密码算法进行扩展的一种自主创新的密码学算法，该算法于 2010 年 12 月 17 日由国家密码管理局设计并发布[51]，可适用于数据的加密、解密、数字签名等操作。SM2 的安全强度比 RSA 高，能够有效防止外部的攻击；其运算效率更高，能够满足大量的数据处理需求。

SM2 的数字签名算法主要包括密钥对生成、签名生成和验证三个环节。

（1）密钥对生成。首先利用椭圆曲线方程：$y^2 = x^3 + ax + b \ mod \ p$，确认曲线，再选择一点 $G(x_g, y_g)$ 作为基点，对曲线做切线、x 对称点运行。用户 A 在[1,n-1]的范围中随机产生一个 d_A 作为自身私钥存储，并通过公式：$P_A = d_A \times G$ 计算出用户 A 的公钥，则用户 A 的密钥对为 (d_A, P_A)。

（2）数字签名的生成。假设需要进行签名的信息为 M，则用户 A 的签名生成运算如下：

1）通过 $H_{256}(Z_A||M)$ 获取消息摘要 e，$Z_A =$

$H_{256}(IDL_A||ID_A||a||b||x_G||y_G||x_A||y_A)$。

2）在[1,n-1]中产生随机数 k, 并计算出椭圆曲线上一点$C_1 = (x_1, y_1) = kG$ 。

3）计算签名参数 $r = (e + x_1)\ mod\ n$ ，若 $r = 0$或 $r + k = n$，则返回（2）重新计算。

4)计算签名参数 $s = \left((1 + d_A)^{-1} \times (k - r \times d_A)\right)\ mod\ n$,若$s = 0$,则返回(2）重新计算。

5）获得消息 M 最终的签名为(r, s)。

（3)签名验签。当验证方用户 B 收到用户 A 发送的消息 M 和签名$(r' + s')$ 时，具体验证步骤如下：

1）判断r' 和 s' 是否在[1,n-1]的范围内且$r' + s' \neq n$ 是否成立，若不成立则验证失败。

2）计算$t = (r' + s')$ ，若$t = 0$，则验证失败。

3）计算$(x_1', y_1') = s'G + tP$, $R = (e' + x_1')\ mod\ n$ ，若 $R = r'$，则验证通过，否则失败。

2.3 供应链需求量预测相关技术

2.3.1 长短时记忆算法的基础理论

长短时记忆（LSTM）神经网络模型是 HOCHREITER 等人[52]基于循环神经网络的优化而提出的，它的出现主要是为了解决循环神经网络在训练时常出现的梯度爆炸问题[53]。由于 LSTM 模型在循环神经网络结构上添加了“门”的概念，使数据能够通过门控结构进行存储和传输，一定程度上提高了数据处理和训练模型的效率。其“门”结构主要包括遗忘门、输入门和输出门。

遗忘门表示在前一时刻有多少信息需要被遗忘，有多少信息需要被当前保留。遗忘门是一个简单的 BP 神经网络，可通过设置权重和偏置对上一时刻的状态进行计算，并利用 $sigmoid$ 函数获得输出值 $f_t \in [0, 1]$，其中 0 表示完全遗忘，1 表示完全保留。可利用公式：$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f)$计算遗忘门的输出。其中，$W_f$和$b_f$分别表示权重矩阵和偏置矩阵，$x_t$表示当前时刻输入值，$h_{t-1}$表示上一时刻隐藏层输出，$\sigma$ 表示激活函数。

输入门是用于控制当前输入信息有多少需要存入当前的记忆单元中，它包含了两部分：由 $sigmoid$ 函数决定更新信息 i_t ；通过 $tanh$ 函数决定候选信息 $\widetilde{C_t}$ 。决定哪

些信息需要会被增加至细胞状态可利用 $i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i)$ 和 $\widetilde{C_t} = tanh\,(W_c \times [h_{t-1}, x_t] + b_c)$进行计算并利用 $C_t = f_t \times C_{t-1} + i_t \times \widetilde{C_t}$ 计算出细胞状态 C_t。其中，W_i、W_c 和 b_i、b_c 分别表示权重矩阵和偏置矩阵，$tanh$ 表示激活函数，C_t、C_{t-1} 分别表示当前时刻和前一时刻的状态存储单元。

输出门是用于决定当前时刻有多少信息需要最终输出。先通过 $sigmoid$ 函数对上一时刻的输入 h_{t-1} 和当前时刻细胞的输入 x_t 计算出控制向量 $O_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o)$，再利用 $tanh$ 函数对上一节点得到的新的细胞状态进行放缩并与 O_t 相乘，即可获得最终的细胞输出 $h_t = O_t \times tanh\,(C_t)$。

2.3.2 特征工程

特征工程是将原始数据转换成更便于表达本质特征的模型数据的过程，使这些数据能更好的应用到模型预测中，提高对不可见数据的模型预测精准度。特征工程能提高多种机器学习模型的性能[54]，其主要包括特征提取、特征选择和特征衍生等技术。

特征提取主要是针对研究对象的各类特征属性进行组合和重构的过程，主要方法有主成分分析法（PCA），通过对原始数据进行标准化处理后计算出协方差矩阵和特征值，并获取最大特征值，再将高纬度特征空间转化为行特征空间，以此简化模型并对数据进行压缩。

特征选择是针对研究对象拥有很多不同特征值时使用的，特征类型主要被分为相关特征、无关特征和冗余特征三类。在预测模型的训练中，只有与模型特征相关性强的数据才有益于预测结果的精准，而无关数据只会降低预测的效果。因此，选用有效的数据特征对预测结果的精准度十分重要，而特征选择主要包括四个过程：生成候选的特征子集、评估特征子集、判断是否满足终止条件、验证特征子集是否有效。

特征衍生是在原始数据的基础上进行相关数值和逻辑的运算，以此来提高预测的能力，常见的特征衍生类型包括特征转换和特征组合。例如，在对时间序列进行预测时，原始数据本身的特征类型十分有限，需要结合大量的衍生特征数据来增强模型拟合能力。

2.4 本章小结

本章主要研究基于区块链的电商供应链交易平台所涉及的技术理论研究。首先介

绍了区块链相关的技术，着重阐述了区块链的基础结构、以太坊和智能合约的技术基础及原理。接着介绍了供应链存储交易间的数据处理技术，主要阐述了以太坊区块链的存储结构、IPFS 技术和 SM2 加密算法。最后介绍了供应需求量预测的相关技术，阐述了 LSTM 的基础理论和特征工程技术。

第三章 面向电商供应链的区块链数据存储方案

区块链本身对构建数据共享环境和建立信任机制具有天然的优势。因此利用区块链技术对电商供应链数据进行统一存储，一定程度上保障了数据的安全性和一致性。但由于区块链技术在数据存储时会不间断实行备份机制，使链上数据的存储压力会随着交易量的增多而逐渐增大，导致电商供应链的整体运作效率降低。为解决该问题本章结合 IPFS 技术和 SM2 算法提出一种面向区块链电商供应链的数据存储方案。首先对上链数据进行分类，再利用 IPFS 技术对文件型数据进行转换，接着对上链数据重构，再利用 SM2 算法对重构数据进行加密认证，确保数据的准确性和隐私性，最后通过实验验证得出此方案有效。

3.1 链上数据异构存储设计

为了减少区块链的存储开销，确保区块链网络中只存储供应链的核心数据，因此提出一种基于以太坊区块链和 IPFS 技术的数据异构存储方案。该方案将电商供应链管理的数据分为结构化数据和非结构化文件数据两类，将需要存储的电商供应链数据进行分解。先利用 IPFS 分布式存储机制，将电商供应链中所涉及的商品图片、视频、文档信息等文件型数据进行转化，分散到不同盘中进行数据块存储，并在此基础上计算出唯一的哈希值作为寻址键，用户可根据寻址键获取对应的文件内容。

基于 IPFS 技术对供应链数据的异构模型如图 3.1 所示，依据电商供应链区块链系统链上数据存储节约的原则，将结构化数据直接上链存储，而文件型数据则经过 IPFS 技术处理为 Hash 值后，再联合标识信息上链存储。具体包含以下几个步骤：

1.对传输数据进行分类识别，将结构化数据以 list 的形式整合，对文件型数据标记为 file 类型；

2.将多个 file 数据利用 IPFS 技术进行批量转换，$\{file_1, file_2, \ldots\ldots, file_n\} = \{hash_1, hash_2, \ldots., hash_n\}$；

3.将所有 file 文件转化的 hash 集合与标识数据一一绑定，$list = [\{id_1, file_{hash_1}\}, \{id_2, file_{hash_2}\}, \ldots\ldots, \{id_n, file_{hash_n}\}]$。

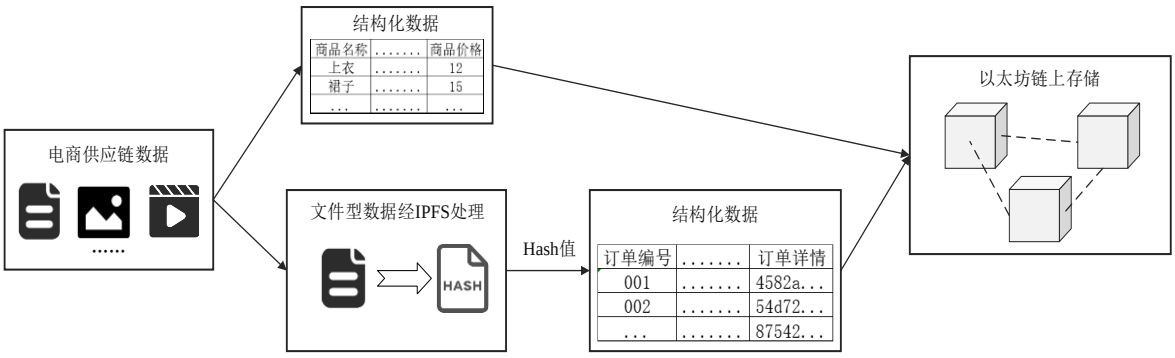

图 3.1 供应链数据的异构模型

对电商供应链数据进行上链操作时，采用 Web3.js 作为前端数据传输的端口，并对接 IPFS 的信息上传接口，使用 Go 语言实现电商供应链数据异构并对接智能合约完成上链，具体逻辑如算法 3.1 所示。

算法 3.1：将供应链数据异构处理并上链

```
1. Begin
2.     Function DataClassification(list)
3.         DataList, FileMap = null
4.         For data in list
5.             If data is file
6.                 FileMap,add(id, file)
7.             Else
8.                 DataList.add(data)
9.         reutrn DataList, FileMap
10.    End Function
11.
12.    Function IPFSHandle(FileMap)
13.        FileHashMap = null
14.        For map in FileMap
15.            hash,err = ipfsAdd(map.file)
16.            FileHashMap.add(map.id, hash)
17.        reutrn FileHashMap
18.    End Function
```

```
19.     Function DataToChain(DataList, FileHashMap)
20.         ParamData = null
21.         For data in DataList
22.             For map in FileHashMap
23.                 If data.id == map.id
24.                     paramData = arrangData(data, map.file_hsah)
25.                     return
26.         tx = saveToChain(paramData)
27.         return    tx
28.     End Function
29. End
```

在设计分类异构数据存储中包含了 3 个子功能：

1. 对上链数据进行分类处理 DataClassification(list)[2 行 - 9 行]。

2. 文件型数据进行 IPFS 转换 IPFSHandle(list)[11 行 - 16 行]。

3. 整合数据上链 DataToChain(list)[18 行 - 26 行]。

子功能中共涉及以下 3 中方法：

1. ipfsAdd(map.file)方法，将文件型数据经过 IPFS 分块机制存储是 IPFS 网络中。

2. arrangData(data, map.file_hsah)方法，根据 ID 标识数据一致性原则将结构化数据与文件 Hash 值整合。

3. saveToChain(paramData)方法，对整合后的数据进行统一上链。

3.2 链下数据同步存储设计

3.1 小节提出基于 IPFS 技术分类异构电商供应链数据存储的方案从源头解决了区块链存储开销较大的问题，但同时也衍生出链上链下存储信息存在差异化的现象，为了避免在电商供应链交易过程中出现信息不一致的情况，结合上链异构存储方案提出链下数据认证同步存储机制。该方案将 SM2 算法融合到异构存储上链机制中，利用 SM2 算法为电商供应链中每个供应商、经销商设定唯一的密钥对，便于他们在电商供应链平台进行交易时对交易信息进行签名和上链。为了提高数据的可用性和实用性，商户可以通过自己的唯一标识和私钥获取并解析链上信息，再通过智能合约对同类型数据进行汇总整合并同步到链下。

3.2.1 异构数据签名机制

异构数据签名上链机制的设定是为了确保上传者身份信息的真实性，以此保证数据的安全性及不可篡改性。针对这种情况，该方案提出利用 2.2.3 小节中 SM2 加密算法的原理对链上异构存储方案进行优化，将经过 IPFS 处理后的异构待上链数据进行公钥加密和私钥签名，提高供应链数据的隐私性。具体的构数据融合 SM2 算法上链流程如图 3.2 所示。

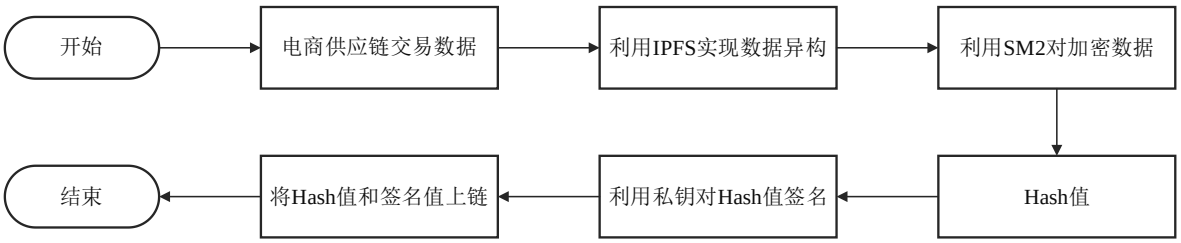

图 3.2 异构数据融合 SM2 算法存储上链流程图

该方案中整体的异构加密流程涉及不同数据结构间的相互转换，将结构化数据转换为体积较小的 Hash 值并将 Hash 值和签名值上链，进一步缓解了链上的存储压力。具体包含以下几个步骤：

1. 获取 IPFS 技术处理并结合结构化数据异构的数据$Data_{new}$；

2. 获取上传者的公钥$pubi_A$和私钥$priv_A$；

3. 对数据将$Data_{new}$进行加密，$Data_{new} = encryption(pubi_A, Data_{new})$；

4. 将上传在的签名值$sign_A$和加密后的数据$Data_{new}$绑定上链，$upload\_chain(sign_A, Data_{new})$（$sign_A$和$Data_{new}$均为一串 Hash 值）。

该方案的具体实施涉及到 SM2 算法的实现及整个供应链数据的隐私性保护，因此利用 GO 语言对整个上链异构存储方案进行优化改进，具体的逻辑算法如 3.2 所示。

算法 3.2：将异构数据加密并进行签名

```
1. Begin
2.     // 获取用户密钥对并存与 IPFS
3.     Function GetUserSM2Key(id)
4.         user = GetUser(id)
5.         userPriKey = GenerateSM2Key(random, user)
6.         userPubKey = GetPubSM2Key(userPriKey)
```

```
7.              keyFile = SaveUserKey(user,userPriKey,userPubKey)
8.              key = UploadIpfs(keyFile)
9.              return key,user
10.     End Function
11.
12.     // 利用公钥给数据加密
13.     Function IsomericEncryption(_data,userPubKey)
14.             data = bytes(_data)
15.             encryptedData = EncryptWithSM2(data,userPubKey)
16.             return Hash(encryptedData)
17.     End Function
18.
19.     // 利用私钥对信息签名
20.     Function HashSign(_msg,userPriKey)
21.             sendMsg = bytes(_msg)
22.             sign = Signature(sendMsg,userPriKey)
23.             return sign
24.     End Function
25. End
```

在设计异构数据加密存储时包含了 3 个子功能：

1. 获取用户的密钥对 GetUserSM2Key(id)[3 行 - 10 行]。

2. 加密异构数据 IsomericEncryption(_data, userPubKey)[13 行 - 17 行]。

3. 对数据签名 HashSign(_msg, userPriKey)[20 行 - 24 行]。

子功能中共涉及了以下 7 种方法：

1. GetUser(id)方法，根据 ID 获取数据库中对应商户的信息。

2. GenerateSM2Key(random, user)方法，根据商户信息和系统随机数，生成该商户的公钥。

3. GetPubSM2Key(userPriKey)方法，是利用商户的私钥信息生成对应的公钥。

4. SaveUserKey(user, userPriKey, userPubKey)方法，设立一个文件用于存放商户的基本信息和对应的密钥对。

5. UploadIpfs(keyFile)方法，将文件信息存于 IPFS 上统一管理，仅返回对应的 Hash 值和商户绑定存储于系统中。

6. EncryptWithSM2(data, userPubKey)方法，利用商户的公钥对经过处理的异构数据进行加密输出$Hash_{data}$。

7. Signature(sendMsg, userPriKey)方法，将加密后的$Hash_{data}$信息，利用商户的私钥进行签名并返回最终需要上链是签名信息$sign_{msg}$。

3.2.2 链下数据认证同步机制

3.2.1 节中详细阐述了链上异构数据加密存储机制，提高了商户交易数据的隐私性和安全性，一定程度上缓解了区块链网络的存储压力，但面对电商供应链这类需要频繁进行数据访问的情况时，会造成以太坊区块链网络的信息访问过载，从而降低电商供应链平台的运作性能。因此提出了一种链下数据同步认证机制，定期将数据区块链中的数据信息同步到链下存储，这种方式不仅缓解了区块链上的访问压力，同时也保证了数据的真实性。具体链下数据认证同步流程如图 3.3 所示。

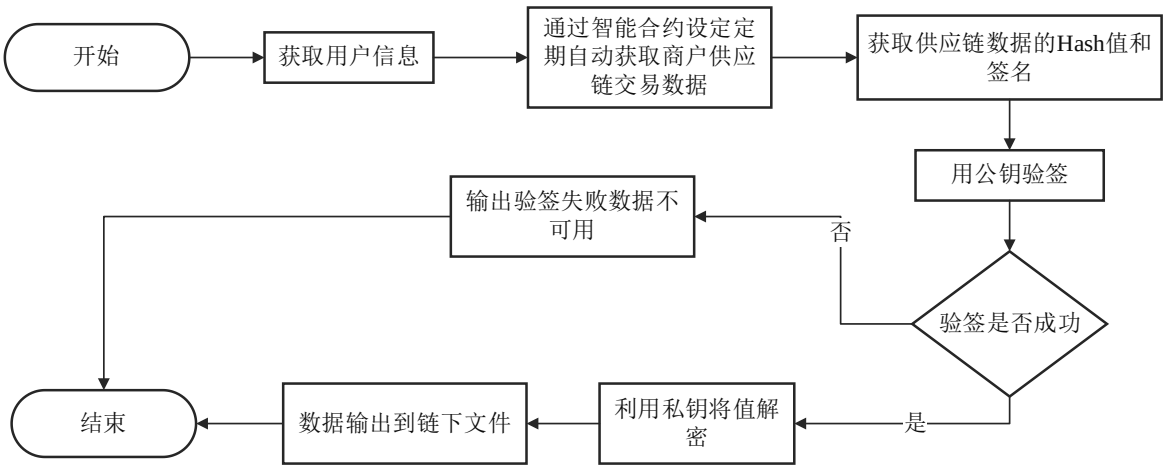

图 3.3 链下数据认证同步流程图

该方案的设计是为保障数据在进行链下同步时，确保信息只有上传者能获取，保护数据的隐私性，避免将隐私数据泄露，具体包括以下几个步骤：

1. 根据用户的唯一标识$User_{id}$通过智能合约获取区块链上对应的供应数据集$HashList = \{sign, [Hash_1, Hash_2, Hash_3, \ldots\ldots, Hash_N]\}$；

2. 获取用户的公钥$pubi_A$进行验签，$flag = verif(sign, pubi_A)$，若$flag = true$，则身份验证成功，否则身份验证失败，程序无法继续进行；

3. 验证成功后，利用用户的私钥$priv_A$对数据进行批量解密，$DataList =$

$[Data_1, Data_2, Data_3, \ldots\ldots, Data_N]$；

4. 将$DataList$批量存入持久型数据库中。

链下数据认证同步方案的实施过程中主要涉及两个部分：一是通过验签确保链上数据的真实性和准确性；二是由于定期同步的数据量较大，手动同步可能会导致系统延时，因此智能合约设置自动化执行参数，让同步机制能在系统使用低峰期自主进行，这种方式不仅避免了人工操作的复杂性也确保了数据的不可篡改性。具体的实现逻辑如算法 3.3 所示。

算法 3.3：数据验签及同步链下存储

```
1. Begin
2.     // 根据用户标识定时获取链上数据
3.     Function TimedAcquisition()
4.         var wg sync.concurrent_execution
5.         For user in userList
6.             wg .AddTime(24:00, GetUserList(user.id))
7.         Wg.start()
8.     End Function
9.
10.    // 智能合约设计：获取用户供应链数据
11.    Function GetUserList(userId)
12.        flag = database.isExist(userId)
13.        If !flag
14.            return error
15.        return hashMap
16.     End Function
17.
18.    // 数据并验签解密
19.    Function DataVerSm2Sig(user, hashMap)
20.        isTrue = VerifySign(user.pub,hashMap.sign)
21.        If !isTrue
```

```
22.                 return error
23.             return Decrypt(user.priv, hashMap.hashList)
24.     End Function
25. End
```

在同步认证机制的核心逻辑算法设计中，主要包括三部分的子功能：

1. 定期通过智能合约上链获取用户的供应链数据的定时器 TimedAcquisition()[3 行 - 8 行]。

2. 智能合约设计，通过用户标识获取供应链数据的执行方法 GetUserList(userId)[11 行 - 16 行]。

3. 验证用户身份并数据进行解密 DataVerSm2Sig(user, hashMap)[19 行 - 25 行]。子功能中共涉及了以下 4 种方法：

1. AddTime(24:00, GetUserList(user.id))方法，在定时器中添加需要并发执行的操作任务。

2. VerifySign(user.pub,hashMap.sign)方法，根据用户公钥对签名信息进行验证，验证通过则进入下一步操作，否则直接返回信息无效。

3. Decrypt(user.priv, hashMap.hashList)方法，借助用户私钥对链上获取的供应链数据进行解密，获取结构化的数据集合 Lsit。

3.3 实验验证与分析

为了验证本课题中链上链下存储方案的有效性，通过模拟供应链交易环境，对大批量的供应链数据上链存储进行存储压力测试。

利用 GO 语言构建一个区块链的数据库环境，模拟多条数据上链存储的过程。在实验中设定了存储 10 条、50 条、100 条、200 条、300 条和 500 条数据存储的情况，分三种情况：原始常规存储、IPFS 技术处理后存储和在 IPFS 处理基础上加密优化后存储，进行存储压力的测试。如图 3.4 所示，蓝色柱状图表示原始数据存储情况，橙色柱状图表示 IPFS 处理后的存储情况，绿色柱状图表示加密优化后的存储情况。保持原数据上链的情况，随着数据量的增多，区块链链上的存储压力也随之增长；经过 IPFS 处理异构后的数据，在上链数据增多时，链上存储压力相对较小；而只存储经过异构方案后加密的 Hash 值，能很大程度上缓解链上存储压力，提升平台整体的存储效率。

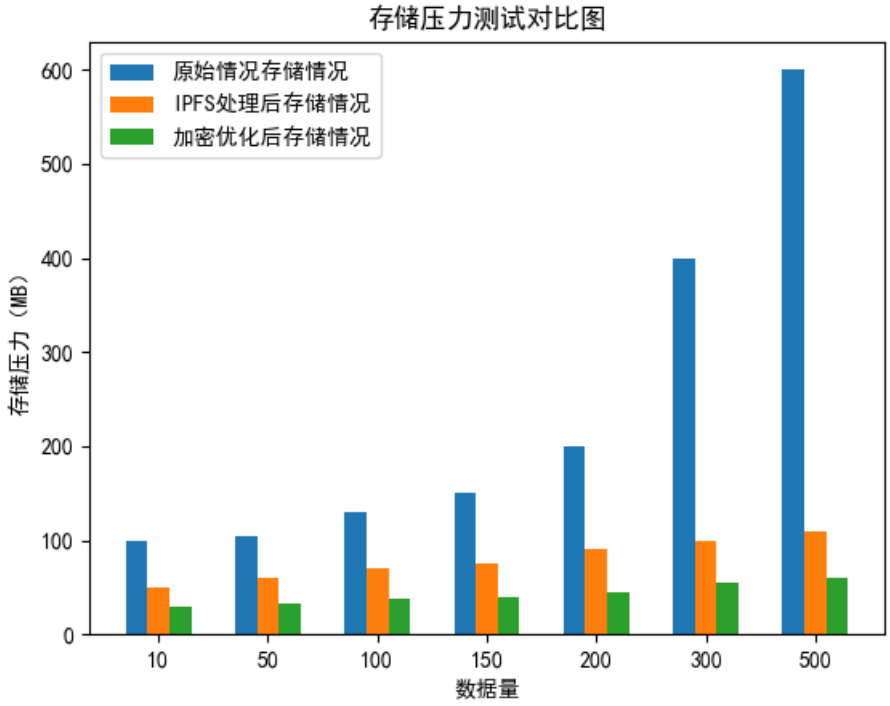

图 3.4 存储压力测试结果图

综上所述，本文提出的链上链下数据存储方案能够有效缓解区块链网络的存储压力，所需存储的数据量越大，该方案的效果越明显。对于电商供应链这类数据交易量非常庞大的业务场景，该方案的实施有助于更好的提升他们的工作效率。

3.4 本章小结

本章为解决区块链链上存储压力提出了链上异构存储方案和链下同步认证机制，首先对上链数据进行分类，将结构化数据和文件型数据分化处理，利用 IPFS 技术对文件型数据进行压缩，再结合 SM2 加密算法对异构数据进行加密并将密文和签名信息上链，其次对链上数据进行链下同步，缩小区块链网络中数据的访问压力。经实验验证该存储方案能有效缓解链上存储压力，以此保存电商供应链平台的高效运行。

第四章 基于区块链的电商供应链需求量预测模型

第三章针对电商供应链数据的特性及问题，提出了基于区块链的链上链下数据同步存储方案，有效缓解了区块链网络中因数据量大而导致的存储过载的问题，同时也提高了电商供应链数据存储的安全性。由于电商供应链业务场景的特殊性，对数据的处理，不仅需要保障其存储环境的安全性，也需要通过对有效数据的合理分析，以提高商户对市场需求的了解及对商品服务的优化。因此，为提高电商供应链数据的使用价值，同时为商户提供精准的运营决策，提出基于区块链的电商供应链需求量预测模型，通过对区块链中供应链交易数据的实时收集与处理，再结合特征工程对 LSTM 算法进行优化后构建需求量预测模型，以提升在区块链中的预测精准度。通过模拟实验证明，该模型的预测性能得到明显改善，不仅提高了区块链中供应链数据的使用率，也帮助商户更好地对商品进行精准化的市场定位。

4.1 基于区块链构建需求量预测模型的总体设计

本文在大数据规模的电商供应链场景下对以太坊区块链进行研究，分析区块链节点间的连接状况、数据交互间的数据分解以及特征识别、提取等技术的实现，并结合以太坊区块链的特性及电商供应链的相关业务流程，研究对电商供应链数据的实时监控，结合神经网络提出分布式模型构架，以解决在传统供应链管理方案中数据信息不准确及管理效率较低等问题，提供可应对异常数据及供应链管理决策的安全管控方案。

根据电商供应链区块链网络管理平台的需求分析，区块链网络能够保障数据的真实有效性，收集处理供应链中的实时交易数据并为不同账户提供权限设置，从而提高电商供应链管理平台的运作效率及安全实用性。与传统供应链管理平台不同的是该平台的设计可以分布式地高效收集整理电商供应链交易中的不同类型数据，并将数据进行统一汇总整理，以此提高整体的运作效率。因此，针对该系统结构的搭建及优化，无需对交易中的实时数据进行特殊处理，只需要通过建立相应的深度学习神经网络预测模型，通过预测模型对区块链中大量的交易数据进行训练，并获取有效的决策分析结果。基于区块链构建的需求量预测模型设计方案如图 4.1 所示，主要

分为数据预处理模块、预测模型构建模块及权限管理模块。

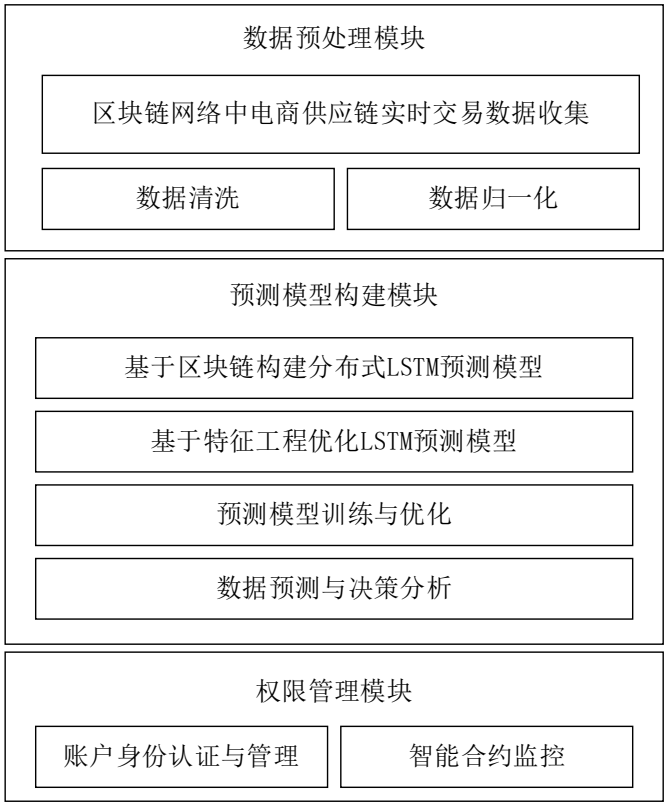

图 4.1 基于区块链的需求量预测模型设计方案

为了提高电商供应链管理的数据安全性及业务运转的高效性，该方案应当满足以下两个需求：

（1）方案能在以太坊区块链中实现数据的实时收集，并确保数据的真实有效性，并对各个节点账户设置管控，避免隐私数据的泄露。

（2）方案能够通过智能合约自动对相应的电商供应链进行分类整合，并结合无监督学习对分布式电商供应链区块链网络数据进行自主预测分析及智能化决策判断，不依赖于人工对数据进行选择和处理，以此实现高效的数据收集和分析，节约各项成本。

4.2 基于以太坊区块链网络进行数据预处理

本文将模拟电商供应链区块链运行数据集作为供应链管理中各业务流程转换中数据监控的测试样本，给数据集包含区块链网络中的节点总数、节点上的存储数据及节点的基本情况的不同维度的网络运行数据，为后续进行神经网络模型的训练、预测等提供统一的操作环境，避免因环境差异而导致的运算误差。在区块链网络中

所获取的数据集不一定能完全符合模型运行时所操作的数据集，因此为了提高样本数据的可用度，避免因数据而导致的模型训练效果不佳等问题，优先利用数据预处理技术将样本数据统一。

对样本数据进行预处理时，先对数据进行分类转换，将文字类型序列经过算法变化转为数字类型序列。本研究中为了适应预测模型的参数设置，选用 LabelEncoder 进行数据的转换处理，具体转换过程如下：

（1）假设原始分类变量 X 中含有 M 个不同的取值，即$X = [x_1, x_2, \ldots, x_m]$；

（2）通过 LabelEncoder 将 M 个不同的值映射为 0 ~ M-1 范围内的不同的整数，即 $[x_1, x_2, \ldots, x_m] \rightarrow [0, 1, \ldots, m-1]$

（3）最后通过统一转换获得对应的整数列编码[0, 1,..., m-1]。

以这种方式能将文字标签有效转化为数字标签，同时保留了文字标签本身的不同属性，也加大了预测模型对数据的学习敏感度，便于模型更好对样本数据进行训练。由于样本数据存在不同指标的数据，造成模型预测精度不准确的问题。因此，为了避免因取值范围过大而造成的问题，在本研究中选用归一化标准来处理所有训练数据，并将数据映射到 0~1 的范围之间，利用公式 $x^* = x - min(x)/max(x) - min(x)$ 计算出规范化的数值，进一步减小对模型性能的影响，实现更精准的数据预测。

4.3 需求量预测模型的构建

4.3.1 长短时记忆算法算法运行原理

长短时记忆算法（LSTM）算法的全称是 Long Short-Term Memory，它是一种特殊的循环神经网络（RNN），该算法主要是用来解决长序列数据的学习和长期依赖问题，它的核心概念包括记忆单元、遗忘门、输入门和输出门。

假设需要对一段长序列 $list = \{a_1, a_2, a_3, a_4, \ldots\ldots, a_i\}$ 进行运算处理，将权重矩阵和偏移矩阵设置为W_f和b_f，输出关键重要的数据，LSTM 算法的主要运算流程如下：

（1）将长序列 list 中的每一个元素列看作一个记忆单元 cell；

（2）通过$sigmoid$函数计算$f_t = \sigma(W_f \times [a_{t-1}, x_t] + b_f)$，决定当前时刻需要从细胞状态中丢弃什么信息，同时为每个单元设置细胞状态；

（3）通过计算$i_t = \sigma(W_f \times [a_{t-1}, x_t] + b_f)$ 、$\widetilde{C_t} = tanh\,(W_f \times [a_{t-1}, x_t] + b_f)$确定需丢弃和更新的数据；

（4）将$f_t \times C_{t-1} + i_t * \widetilde{C_t}$ 即可完成数据细胞状态的更新；

（5）最后通过 $\sigma(W_f \times [a_{t-1}, x_t] + b_f) \times tanh(C_t)$ 获得最终需要输出的序列。

4.3.2 基于区块链构建分布式 LSTM 预测模型

不同于传统的需求量预测模型，本文提出了适用于电商供应链区块链的分布式数据收集处理并进行训练的预测模型架构，如 4.2 所示。

通过获取电商供应链区块链网络中各区域内的节点信息作为分布式场景下的供应链数据交易节点，解决传统网络中心供应链数据差异化的问题。针对全网收集区块链网络中的供应链交易数据，先通过对区域模块下的数据进行统一数据预处理进行转换和规范，再进行汇总整合后做归一化处理，最后根据需求提取后续训练、预测模型在数据分析时所需的特征属性，并基于 LSTM 算法实现电商供应链需求量预测模型。

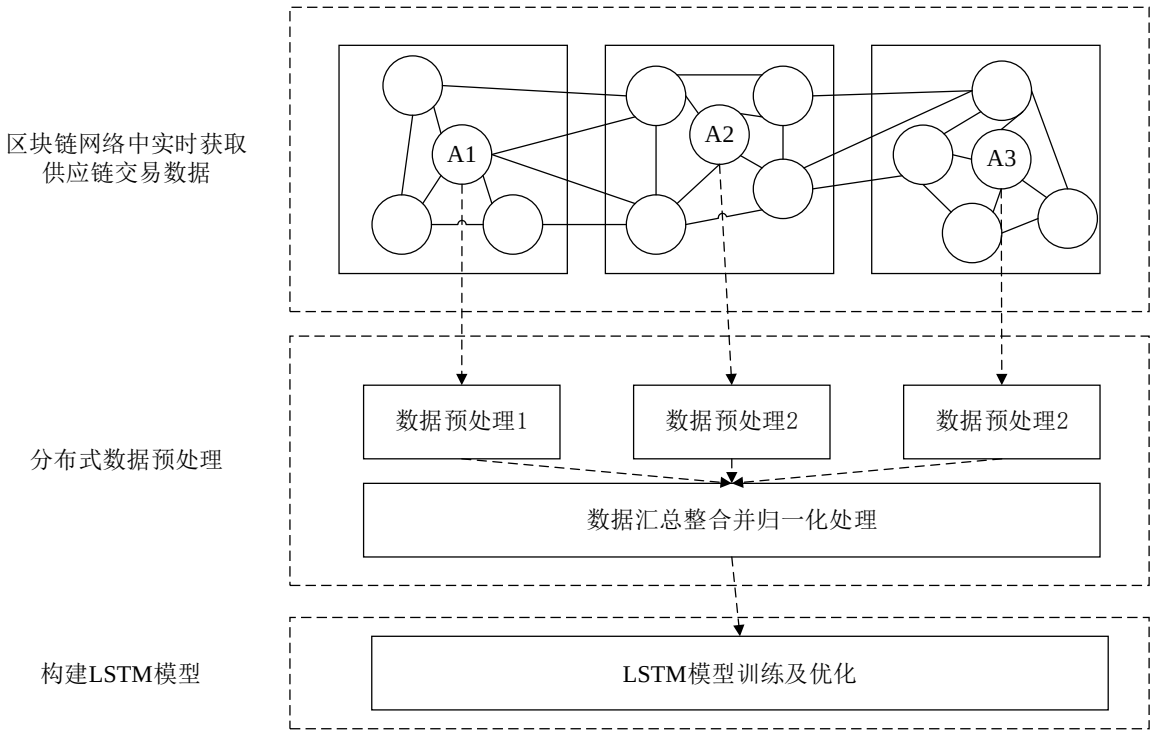

图 4.2 分布式 LSTM 预测模型架构

该分布式需求量预测模型，是基于区块链网络中各节点的数据公开透明性，将供应链各个环节的交易数据进行实时收集，若数据被篡改的，则无法在区块链网络中进行数据传输，以此来保证数据的准确性和安全性。将数据通过分布式收集后，利用数据分类转换和归一化标准，将供应链交易数据统一转换为可便于模型训练的特