

KAUNAS UNIVERSITY OF TECHNOLOGY

ZEBA MAHMOOD

RESEARCH AND IMPLEMENTATION OF
ANOMALY DETECTION METHOD USING
FEDERATED LEARNING AND BLOCKCHAIN
IN IMAGE DATASETS

Doctoral dissertation
Natural Sciences, Informatics (N 009)

2023, Kaunas

This doctoral dissertation was prepared at Kaunas University of Technology, Faculty of Informatics, Department of Software Engineering during the period of 2018–2022.

The doctoral right has been granted to Kaunas University of Technology together with Vilnius Gediminas Technical University and Vytautas Magnus University.

Scientific Supervisor:

Prof. Dr. Vacius JUSAS (Kaunas University of Technology, Natural Sciences, Informatics, N 009)

Edited by: English language editor Dr. Armandas Rumšas (Publishing House *Technologija*), Lithuanian language editor Rozita Znamenskaitė (Publishing House *Technologija*)

Dissertation Defense Board of Informatics Science Field:

Prof. Dr. Hab. Rimantas BARAUSKAS (Kaunas University of Technology, Natural Sciences, Informatics, N 009) – **chairperson**;

Senior Researcher Dr. Jolita BERNATAVIČIENĖ (Vilnius University, Natural Sciences, Informatics, N 009);

Prof. Dr. Gintaras PALUBECKIS (Kaunas University of Technology, Natural Sciences, Informatics, N 009);

Prof. Dr. Agnė PAULAUŠKAITĖ-TARASEVIČIENĖ (Kaunas University of Technology, Natural Sciences, Informatics, N 009);

Prof. Dr. Filippo SANFILIPPO (University of Adger, Norway, Natural Sciences, Informatics, N 009).

The public defense of the dissertation will be held at 9 a.m. on 30 August 2023 at the public meeting of the Dissertation Defense Board of Informatics Science Field in Rectorate Hall at Kaunas University of Technology.

Address: Donelaičio 73-402, Kaunas, LT-44249, Lithuania

Phone: (+370) 608 28 527; email doktorantura@ktu.lt

The doctoral dissertation was sent out on 28 July, 2023.

The doctoral dissertation is available on the internet at <http://ktu.edu>, at the libraries of Kaunas University of Technology (K. Donelaičio 20, Kaunas, LT-44239, Lithuania), Vytautas Magnus University (K. Donelaičio 52, Kaunas, LT-44244, Lithuania) and Vilnius Gediminas Technical University (Saulėtekio 14, Vilnius, LT-10223, Lithuania).

KAUNO TECHNOLOGIJOS UNIVERSITETAS

ZEBA MAHMOOD

ANOMALIJŲ NUSTATYMO METODO
TYRIMAS IR ĮGYVENDINIMAS VAIZDO
DUOMENŲ RINKINIUOSE TAIKANT
FEDERACINĮ MOKYMĄ IR BLOKŲ
GRANDINIŲ TECHNOLOGIJĄ

Daktaro disertacija
Gamtos mokslai, informatika (N 009)

2023, Kaunas

Disertacija rengta 2018–2022 metais Kauno technologijos universiteto Informatikos fakultete, Programinės įrangos inžinerijos katedroje.

Doktorantūros teisė Kauno technologijos universitetui suteikta kartu su Vilniaus Gedimino technikos universitetu ir Vytauto Didžiojo universitetu.

Mokslinis vadovas:

prof. dr. Vacius JUSAS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009)

Redagavo: anglų kalbos redaktorius dr. Armandas Rumšas (leidykla „Technologija“), lietuvių kalbos redaktorė Rozita Znamenskaitė (leidykla „Technologija“)

Informatikos mokslo krypties disertacijos gynimo taryba:

prof. habil. dr. Rimantas BARAUSKAS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009) – **pirmininkas**;

vyresn. m. d., dr. Jolita BERNATAVIČIENĖ (Vilniaus universitetas, gamtos mokslai, informatika, N 009);

prof. dr. Gintaras PALUBECKIS (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009);

prof. dr. Agnė PAULAUŠKAITĖ-TARASEVIČIENĖ (Kauno technologijos universitetas, gamtos mokslai, informatika, N 009);

prof. dr. Filippo SANFILIPPO (Adgerio universitetas, Norvegija, gamtos mokslai, informatika, N 009).

Disertacija bus ginama viešame Informatikos mokslo krypties disertacijos gynimo tarybos posėdyje 2023 m. rugpjūčio 30 d., 9 val., Kauno technologijos universiteto Rektorato salėje.

Adresas: K. Donelaičio g. 73-402, Kaunas, LT-44249, Lietuva

Tel. (+370) 608 28 527; el. paštas doktorantura@ktu.lt

Disertacija išsiųsta 2023 m. liepos 28 d.

Su disertacija galima susipažinti interneto svetainėje <http://ktu.edu>, Kauno technologijos universiteto (K. Donelaičio g. 20, Kaunas, LT-44239, Lietuva), Vilniaus Gedimino technikos universiteto (Saulėtekio al. 14, Vilnius, LT-10223, Lietuva) bei Vytauto Didžiojo universiteto (K. Donelaičio g. 52, Kaunas, LT-44244, Lietuva) bibliotekose.

© Z. Mahmood, 2023

CONTENT

LIST OF TABLES.....	7
LIST OF FIGURES.....	8
ABBREVIATIONS.....	10
1.1 I TRODUCTION.....	11
1.1 Motivation.....	12
1.2 Object of the Research.....	12
1.3 Aim and Objectives.....	12
1.4 Research Methodology.....	13
1.5 Defended Statements.....	13
1.6 Scientific Novelty.....	14
1.7 Dissertation Structure.....	14
2.1 OVERVIEW.....	15
2.2 BLOCKCHAIN.....	15
2.2.1 Architecture of Blockchain.....	17
2.2.2 Smart Contract.....	19
2.2.3 Zero-Knowledge Proof.....	20
2.2.4 Consensus Mechanisms.....	21
2.3 FEDERATED LEARNING (FL).....	21
2.3.1 Federating Learning and Blockchain.....	23
2.3.2 Industrial Applications of Blockchain and Federated Learning.....	25
2.4 ANOMALY DETECTION.....	27
2.4.1 Traditional Ways of Anomaly Detection.....	28
2.4.2 Supervised Learning Approach.....	29
2.4.3 Unsupervised Learning Approach.....	30
2.4.4 Support Vector Machine.....	31
2.4.5 Clustering-Based Methods.....	32
2.4.6 Autoencoder.....	32
2.4.7 Recurrent Neural Network.....	33
2.4.8 Anomaly Detection with Federated Learning and Blockchain.....	34
2.5 SECTION SUMMARY.....	40
3. RESEARCH DESIGN AND METHODS.....	40
3.1 OVERVIEW.....	40
3.2 DATA COLLECTION.....	41
3.2.1 MNIST.....	41
3.2.2 E-MNIST.....	42
3.2.3 Fashion-MNIST.....	43
3.2.3 MNIST-C.....	44
3.2.4 K-MNIST.....	45
3.3 Data Pre-Processing.....	47
3.4 FEDERATED LEARNING.....	48
3.5 ANOMALY DETECTION SYSTEM.....	51

3.5.1 Threshold Calculation.....	51
3.5.2 Support Vector Machine.....	52
3.5.3 Autoencoder.....	55
3.6 ZERO-KNOWLEDGE PROOF.....	57
3.7 BLOCKCHAIN.....	58
3.8 SUPERVISED LEARNING FRAMEWORK.....	58
3.9 UNSUPERVISED LEARNING FRAMEWORK.....	61
3.10 APPLICATION FRAMEWORK.....	64
3.11 SECTION SUMMARY.....	68
4.1 OVERVIEW.....	69
4.2 EXPERIMENTAL SETUP.....	69
4.3 SUPERVISED LEARNING APPROACH TO DETECT ANOMALIES BY USING ZERO-KNOWLEDGE PROOF ALONG WITH FEDERATED LEARNING AND BLOCKCHAIN.....	70
4.3.1 Experimental Steps.....	70
4.3.2 Results.....	75
4.3.2.1 Fashion-MNIST Dataset.....	76
4.3.2.2 MNIST Dataset.....	79
4.3.2.3 Supervised E-MNIST Dataset.....	81
4.4 UNSUPERVISED LEARNING APPROACH TO DETECT ANOMALIES BY USING ZERO-KNOWLEDGE PROOF ALONG WITH FEDERATED LEARNING AND BLOCKCHAIN.....	84
4.4.1 Fashion-MNIST Dataset.....	91
4.4.1.1 Parameters and Layers.....	91
4.4.1.2 Results.....	92
4.4.2 MNIST Dataset.....	95
4.4.2.1 Parameters and Layers.....	95
4.4.2.2 Results.....	96
4.4.3 E-MNIST Dataset.....	99
4.4.4 MNIST-C Dataset.....	100
4.4.5 K-MNIST Dataset.....	105
4.5 SECTION SUMMARY.....	110
5.CONCLUSION.....	111
SANTRAUKA.....	113
LITERATURE.....	156
CURRICULUM VITAE.....	165
LIST OF PUBLICATION.....	165
ACKNOWLEDGEMENTS.....	167

LIST OF TABLES

Table 1: Research Methodology.....	13
Table 2: Smart contracts comparison with multiple platforms.....	20
Table 3: Zero-knowledge proof Descriptive Analysis.....	21
Table 4: Comparison table between Federated Learning and Blockchain.....	24
Table 5: Blockchain and Federated Learning Implementation in Industrial Sectors.....	26
Table 6: Methods of Supervised learning for anomaly detection.....	30
Table 7: Methods of Unsupervised learning for anomaly detection.....	31
Table 8: Anomaly detection using different techniques.....	36
Table 9: Module Explanation for Anomaly detection using ZKP, FL and blockchain using Supervised and Unsupervised approaches.....	67
Table 10: Experimental environment seup.....	69
Table 11: ZKP validation.....	72
Table 12: Blockchain verifiable reords.....	73
Table 13: Indexes of anomalies in the Fashion-MNIST dataset.....	77
Table 14: Model test accuracies results for MNIST dataset.....	80
Table 15: Values of losses per Epoch using Supervised Learning Approach for E-MNIST dataset.....	82
Table 16: Model test results for E-MNIST dataset	83
Table 17: Un-Supervised Blockchain Transactions values for Fashion- MNIST dataset.....	91
Table 18: Un-supervised trained and non-trainable parameters values for Fashion-MNIST dataset.....	92
Table 19: Reduction of losses as per Epoch in un-supervised Fashion-MNIST dataset.....	92
Table 20: Trained epochs values for Confusion matrix in Fashion-MNIST dataset.....	94
Table 21: Trained epochs value Evaluation metrics for Fashion-MNIST dataset.....	94
Table 22: Parameters model predictions for MNIST dataset.....	96
Table 23: Trained epochs value by Confusion matrix for MNIST dataset.....	97
Table 24: Trained epochs value Evaluation metrics for MNIST dataset.....	97
Table 25: Epochs loss values for E-MNIST dataset.....	99
Table 26: Trained epochs value Evaluation metrics for E-MNIST dataset.....	100
Table 27: Predicted Evaluation Metrics for MNIST-C dataset.....	102
Table 28: Predication model performance values for MNIST-C dataset.....	105
Table 29: Epoch's loss values for K-MNIST dataset.....	106
Table 30: predicted Evaluation Metrics for K-MNIST dataset.....	107

LIST OF FIGURES

Figure 1. Example of Blockchain (Z. Zheng et al., 2017)	17
Figure 2. Baran, P. (1964). On Distributed Communication, Memorandum RM-3420-PR.....	17
Figure 3. Block Architecture Representation.....	18
Figure 4. Transaction process using Federated Learning.....	23
Figure 5. The Architecture of Blockchain Based Federated Learning.....	25
Figure 6: Data Collection Visualization for MNIST dataset.....	42
Figure 7: Data Collection Visualization for EMNIST dataset.....	43
Figure 8: Data Collection Visualization for Fashion- MNIST dataset.....	44
Figure 9: Data Collection Visualization for MNIST-C dataset.....	45
Figure 10: Data Collection Visualization for K-MNIST dataset.....	46
Figure 11: Processing method for Data Pre-Processing	47
Figure 12: The Anomaly Detection process using FL Trained Model.....	49
Figure 13: Retraining Process of Federated Learning.....	50
Figure 14: SVM separates classes of MNIST dataset using clear boundaries.....	52
Figure 15: SVM separates classes of EMNIST dataset using clear boundaries.....	53
Figure 16: SVM separates classes of MNIST-C dataset using clear boundaries.....	53
Figure 17: SVM separates classes of Fashion-MNIST dataset using clear boundaries.....	54
Figure 18: SVM separates classes for K-MNIST dataset using clear boundaries....	54
Figure 19: Methods for Anomaly detection for Auto Encoder and SVM.....	55
Figure 20: Supervised Learning approach using Federated learning, Blockchain and ZKP.....	61
Figure 21: Un-Supervised Learning approach using Federated learning, Blockchain and ZK.....	63
Figure 22: Proposed framework basic components.....	65
Figure 23: Framework for Anomaly detection using ZKP, FL and blockchain using. Supervised and Unsupervised approaches.....	66
Figure 24: Supervised Learning Approach using ZKP and FL along with Blockchain.....	75
Figure 25: Anomalous images in Fashion-MNIST Dataset.....	76
Figure 26: Anomalies detected in Fashion-MNIST dataset.....	78

Figure 27: Locally trained Model Losses in Local batch for Fashion-MNIST dataset.....	79
Figure 28: Locally trained Model Losses per round for Fashion-MNIST	79
Figure 29: Federated trained model using MNIST dataset.....	79
Figure 30: Anomalies detected in MNSIT dataset.....	81
Figure 31: Supervised batch losses using MNIST dataset.....	81
Figure 32: Trained model using federated learning for E-MNIST dataset.....	81
Figure 33: Supervised batch losses using E-MNIST dataset.....	83
Figure 34: Supervised Losses per round using E-MNIST dataset.....	83
Figure 35: Detected Anomalies in E-MNIST dataset	84
Figure 36: Un-supervised Learning approach Using ZKP, FL and Blockchain....	86
Figure 37: Un-supervised learning approach Clustering.....	89
Figure 38: Un-Supervised layers for Fashion-MNIST dataset.....	91
Figure 39: Predicted precision score Confusion matrix for Fashion-MNIST dataset.....	93
Figure 40: Anomaly detected predicted plot in Fashion-MNIST dataset.....	95
Figure 41: Un-Supervised layers for MNIST dataset.....	95
Figure 42: Predicted precision score Confusion matrix for MNIST dataset.....	98
Figure 43: Detected anomalies predicted plot in MNIST dataset.....	98
Figure 44: Detected anomalies predicted plot in E-MNIST dataset.....	100
Figure 45: Anomaly detection by multiple anomalous behaviors.....	101
Figure 46: un-supervised batch losses using MNIST-C dataset.....	101
Figure 47: un-supervised batch accuracy using MNIST-C dataset.....	102
Figure 48: Forecasted plot of the results of anomaly detection using the MNIST-C dataset.....	103
Figure 49: Un-supervised batch losses using MNIST-C dataset.....	104
Figure 50: Un-supervised batch accuracy using MNIST-C dataset.....	104
Figure 51: MNIST-C Model accuracy plot.....	105
Figure 52: K-MNIST Model accuracy plot.....	107
Figure 53: Predicted precision score Confusion matrix for K- MNIST dataset..	108
Figure 54: un-supervised batch losses using MNIST-K dataset.....	109
Figure 55: un-supervised batch accuracy using MNIST-K dataset.....	109
Figure 56: Sample images in K-MNIST dataset.....	109
Figure 57: Anomalies detection in K-MNIST dataset.....	109

LIST OF ABBREVIATIONS AND DEFINITIONS

AI – Artificial Intelligence
BCFL – Blockchain Based Federated Learning
BIC – Bayesian Information Criterion
CFT – Crash Fault Tolerant
CNNs – Convolutional neural networks
DAGs – Directed acyclic graph
dApps – Decentralized application
DoS – Denial-of-Service
EOS – Electro-Optical System
FedAvg – Federated Averaging
FedMA – Federated Matched Averaging
FL – Federated Learning
FTL – Federated transfer learning
GDPR – General Data Protection Regulation
GPU – Graphics processing unit
HFL – Horizontal federated learning
LSTM – Long Short-Term Memory network
ML – Machine learning
MSELoss – Mean Squared loss
NN- Neural Network
Non-FL – Non-Federated Learning
non-IID – Non-independent and identically distributed
P2P – Peer-to-peer
PCa – Principal Component Analysis
PoA – Proof of Authority
PoC – Proof of Concept
PoI – Proof of Identity
PoS – Proof-of-stake
PoW – Proof of Work
RNN – Replicator neural network
SGD – Stochastic gradient descent
SPOF – Framework avoids single points of failure.
SVM – Support vector machine
VFL – Vertical federated learning

INTRODUCTION

In response to the growing number of digital image datasets and the complexity of cyber threats, it has become essential to develop new techniques for detecting anomalies. With the increasing importance of digital data in today's world, it is crucial to continue researching and developing effective anomaly detection methods to ensure the security and privacy of individuals and organizations. Blockchain technology can also be used to regulate the distribution of data in addition to federated learning. The intended use of this thesis is to develop a framework for the detection of image dataset anomalies which integrates the capabilities of the blockchain technology and federated learning. This will enable us to enhance the precision of our techniques and protect the security and confidentiality of our data. Multiple entities can collaborate to train models via federated learning, a decentralized approach that eliminates the need to share data. The detection of anomalies in image datasets is a challenging task requiring accurate and efficient methods. To address this issue, we worked on different approaches, including supervised and unsupervised learning, as well as *Federated Learning* (FL) and blockchain technologies. In addition, zero knowledge proof has emerged as a potential solution for enhancing data privacy and security in anomaly detection. The training and detection stages are crucial to establish the detected occurrences and evaluate the test findings. Since a blockchain transaction cannot be changed or deleted by anyone, the structure's ability to resist tampering is a sign of its robustness. Moreover, it demonstrates that transactions initiated with missing private keys cannot be corrected. With the advancement in technology, various approaches have been developed to detect anomalies with high accuracy. In this work, we shall discuss how *Federated Learning* (FL), Blockchain, and zero-knowledge proof can be used for detecting anomalies in image datasets using supervised and unsupervised learning approaches. These techniques help to identify both known and unknown anomalies and adapt to changing environments.

Keywords: Blockchain, Federated Learning, Machine Learning, Supervised Learning, Unsupervised Learning, Anomaly Detection, Blockchain Based Federated Learning, Zero-Knowledge Proof.

1.1. Motivation

In this work, it was decided to develop a new abstraction-based approach for anomaly detection using federated learning and blockchain. The first reason was that the different existing approaches, which had been addressed by many researchers or authors, did not address the primary issues in image datasets. Detecting anomalies usually requires centralized data storage, which raises privacy and security concerns. Another reason is that literature studies show that there are no available approaches towards the reconstruction of timelines that are implemented on multiple platforms. These methods may not be able to handle large and complex datasets, which results in reduced accuracy. Third, they may not be able to detect anomalies in real-time, thus they are made less suitable for time-sensitive applications. Therefore, the proposed approach aims to overcome these limitations by utilizing federated learning and blockchain technology to enable decentralized and secure anomaly detection in image datasets. By leveraging the power of multiple devices and platforms, we undertake to handle large and complex image datasets while maintaining data privacy and security. Additionally, real-time anomaly detection is possible while making it suitable for time-sensitive applications.

1.2 Object of the Research

The object of this research is the development of the mechanism applied to investigate Anomaly detection by using Zero-Knowledge proof using Federated Learning and Blockchain.

1.3. Aim and Objectives

The aim is to improve anomaly detection novel methods in images datasets using Federated Learning and Blockchain.

Objectives of the thesis:

- To create and deploy a federated learning algorithm capable of extracting knowledge from image datasets.
- To enhance the accuracy and efficiency of anomaly identification in image datasets compared to the current state-of-the-art achievements.
- To Integrate the Federated Learning algorithm with the blockchain-based data sharing and storage system to enable decentralized and secure image dataset anomaly detection.
- To evaluate the performance and accuracy of the proposed framework.

1.4 Research Methodology

Table 1. Research Methodology

1. Analysis of the previous work and Research Work.	Analysis of blockchain and Federated Learning in the context of threats prevention. Different studies about the effectiveness of the trained model (Federated Learning) and the genesis of blockchain. Analysis of robustness of the performed studies for anomaly detection.
2. Literature work for standard technologies and privacy preservation mechanism approaches against threats.	Existing work for anomaly detection for standard technologies against privacy protection abstractive approaches in various mediums of threats.
3. Research and design of Methods used for Anomaly detection.	Data pre-processing, FL models, and implementation of proposed work included techniques for anomaly detection.
4. Experimental values and novelty to deduct anomaly threats.	Results of experimental values and validated algorithm against anomaly deduction.
5. Implementation and Analysis of results.	The implementation of the proposed FL model and approaches as literature studies show the importance of enhancement in processes.

1.5 Defended Statements

The statements defended by the dissertation are the following:

1. The Blockchain-based FL model allows to determine whether the image is normal, or the image has an anomaly.
2. The proposed approach enables to determine anomaly either in supervised or unsupervised way depending on the image dataset.

3. The novel approach allows to conduct the analysis in image datasets securely and to decentralize the data of image datasets, which reduces the risk of data breaches and ensures data privacy through blockchain-based data storage.

1.6 Scientific Novelty

1. This thesis proposes a novel approach of anomaly detection in image datasets by using Federated Learning. This enables learning from distributed data sources while preserving privacy and security.
2. The thesis introduces a unique blockchain-based system for securely sharing and storing sensitive image data. The system incorporates *Zero Knowledge Proof* (ZKP) to further enhance the security and privacy of the data.
3. The proposed framework combines Federated Learning and blockchain to enable decentralized, secure, and privacy-preserving anomaly detection in image datasets.

1.7 Dissertation Structure

The dissertation is divided into five chapters:

1. The first chapter introduces the reader to the aim of the thesis and the topic of anomaly detection method using federated learning and blockchain.
2. The second chapter analyzes the currently available literature and several feature extraction methods along with the traditional machine learning approaches and federated learning, including blockchain and attacks against anomaly detection.
3. The third chapter introduces the methods for federated learning model and training model for pre-processing data.
4. The fourth chapter introduces the Experimental structure and provides all data collection result values.
5. The final section summarizes the whole work and gives main conclusions for this thesis along with recommendations for further research.

2. STATE OF THE ART ANALYSIS

2.1 Overview

This section explains anomaly detection strategies in detail. The section delves into a variety of approaches, such as *Zero Knowledge Proof* (ZKP) computations, supervised and unsupervised learning algorithms, and clustering-based methods such as K-means, *Support Vector Machines* (SVM), and Autoencoders. It also emphasizes on blockchain architecture and its different models, such as centralized, decentralized, and distributed. Moreover, the crucial role of smart contracts in automating contracts and improving transparency and security is highlighted. The concept of zero-knowledge proof is investigated, with an emphasis on its capacity to validate statements without revealing sensitive information. This section also examines such consensus mechanisms as Proof of Work and Proof of Stake, by emphasizing the relevance of these processes in maintaining the transaction legitimacy and network security. This section also goes into the use of blockchain technology and *Federated Learning* (FL) in anomaly detection systems. Ongoing research focuses on overcoming difficulties, enhancing accuracy, efficiency, and scalability, and investigating novel techniques to improve anomaly detection skills.

2.2 Blockchain

Blockchain technology has emerged as a revolutionary concept that has been garnering significant attention across various industries. Initially introduced in 2008 by Satoshi Nakamoto in his whitepaper on Bitcoin, blockchain has evolved beyond cryptocurrencies to become a disruptive force in multiple sectors [1]. Blockchain technology offers numerous benefits that have contributed to its widespread adoption and potential for transformative impact across various industries. Decentralization is a key feature which eliminates the need for intermediaries, improves efficiency, and increases trust among participants [2]. The transparency of the blockchain, facilitated by the immutable and publicly accessible ledger, enhances accountability and reduces the risk of fraud or manipulation [3]. The advanced cryptographic techniques employed in blockchain ensure the security and integrity of data, while the decentralized nature of the technology makes it more resilient to attacks [4]. Immutability is another advantage of blockchain, as once a transaction has been recorded, it becomes nearly impossible to alter, thereby making it suitable for secure and tamper-resistant record-keeping [5]. Blockchain's efficiency and cost savings arise from its ability to automate processes through smart contracts, reducing the time and costs associated with manual processing and reconciliation [6]. The traceability and auditability provided by blockchain are valuable in supply chain management, where stakeholders can track the movement of goods and verify their authenticity [7]. Blockchain technology has the potential to improve financial services by increasing access and financial inclusion [8]. It

enables individuals to have secure and direct control over their financial transactions while bypassing the traditional intermediaries and reducing barriers to financial inclusion [9]. The interoperability of blockchain allows for seamless integration and sharing of data, assets, and processes across different systems and organizations, fostering collaboration and innovation [10]. Data privacy is enhanced through such techniques as zero-knowledge proof and private transactions which provide individuals with control over their personal data while maintaining transparency and trust [11]. Blockchain's decentralized and transparent nature fosters trust among the participants by eliminating the need for intermediaries and providing a tamper-proof record of transactions [12]. These benefits of blockchain technology have the potential to revolutionize industries, streamline processes, and create new opportunities for innovation and collaboration. However, it is crucial to consider the specific case of use, technical challenges, and regulatory considerations when implementing blockchain solutions [13].

A virtual currency system eschewed a central authority for issuing currency, transferring ownership, and confirming transactions. Satoshi Nakamoto introduced blockchain in 2008 with his whitepaper on Bitcoin, the first peer-to-peer (P2P) cryptocurrency [14]. To facilitate digital transactions, the technology combined cryptography, a concept that was already well known in computer science at the time, with a novel Merkle tree data structure. Cryptocurrencies became effective not only for economics industries but all sectors including governmental [15] and Private Industrial technology needs [16]. Ralph C. Merkle's [17] work published in the book *The Evolution of Merkle's Tree Structure*, regarding the evolution of the hash tree structure (first published in 1980) depicts how the clever cryptographic structure makes use of layered hash functions which record many transactions in a safe and unchangeable manner within the block of the chain. Blockchain contains elements of Distributed Ledger, Smart Contracts, Cyber Security and Privacy Preservation. In recent years, many new directions came to the limelight while highlighting the blockchain aspects for security and privacy. By simplifying blockchain as a multitasking entity, we can use blockchain and modify it according to the requirements using the Consensus algorithm.

Fig. 1. presents an example of a blockchain which includes the detailed description of hashing and a timestamp from genesis block Cryptocurrencies, especially Bitcoin, associated with the *Blockchain 1.0* technology [18]. Blockchain can be used to provide incentives to encourage community participation in FL. Blockchain also provides traceability, transparency, and the associated digital trust in FL models, in addition to incentives [19].

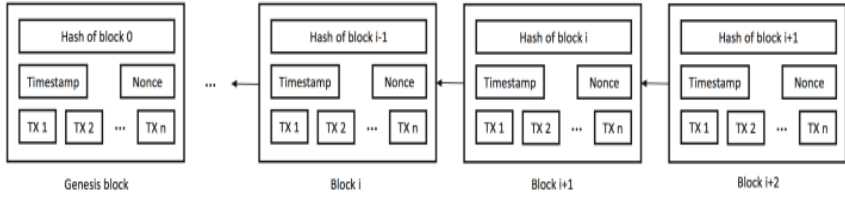


Fig. 1. Example of Blockchain (Z. Zheng *et al.*, 2017)

2.2.1 Architecture of Blockchain

The basic blockchain architecture is based on centralized [20], decentralization [21], distributed [22] structures. **Fig. 2.** illustrates distributed communication.

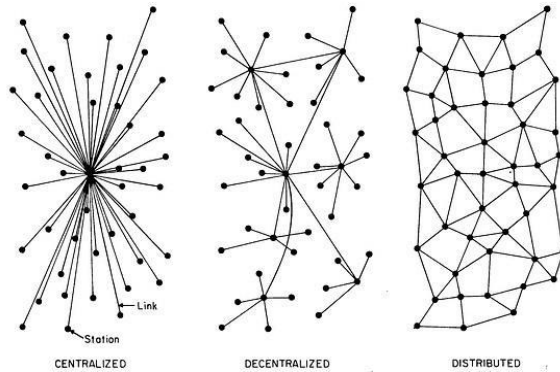


Fig. 2. Distributed Communication, Memorandum RM-3420-PR (Baran, P. (1964))

i. Centralized

Centralized blockchains are a type of blockchain where a single entity or a group of entities control the network's nodes. In contrast to decentralized blockchains, centralized blockchains have a centralized governance structure, which means that decisions related to the network's development and maintenance are made by a central authority. Social networking websites, such as Facebook, Twitter, and Quora, are examples of centralized systems. These platforms have a centralized control over the user data and the content that is shared on them.

ii. Decentralized

Decentralization servers are distributed all over the network instead of being controlled by one server. Decentralized blockchains are a type of blockchain technology which operates on a peer-to-peer network without any central authority or control. Decentralization is a fundamental characteristic of blockchain technology because it ensures that the network is transparent, secure, and immutable. Decentralized blockchains offer several advantages over centralized blockchains. There have been Dapps introduced in a wide range of fields over the past few years, including finance, online games, and trading exchanges. Decentralization has evolved from network engineering [23], to software engineering [24], and then into information, management, and organizational sciences [25], [26].

iii. Distributed

A ledger with distributed nodes records, shares and synchronizes transactions instead of keeping them in a centralized location. A distributed ledger is a database which is accessible by multiple participants across multiple locations and geographies.

- The distributed ledger technology eliminates the need for a central authority to guard against manipulation.
- Distributed ledgers are public ledgers of transactions and contracts that are maintained in different locations and by different people.

For building blockchain architecture, we require nodes and blocks so that every node in the pool will confirm a transaction, and, based on their collective knowledge of consensus, will add up to a new block in the chain as the block structure presented in **Fig. 3**. Blockchain also uses Proof of Work, which is the consensus mechanism, to allow nodes to do work and verify transactions.

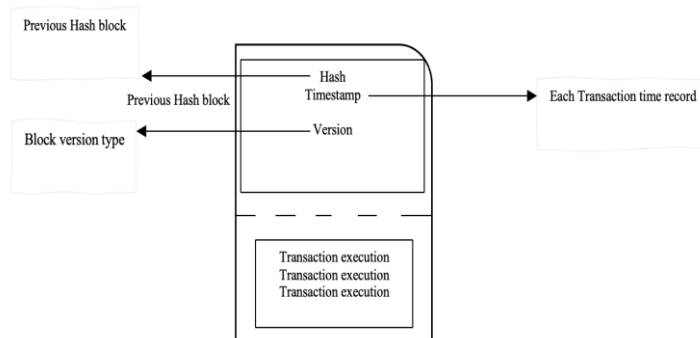


Fig. 3. Block Architecture Representation

The concept of blockchain can be explained in simple terms as blocks attached

to chains. Information in each block can be stored. Each block contains a Merkle tree [27], which calculates every hash value from the transaction information. As a result of this scheme, neither the transaction data in each block, nor the blocks that are linked in the blockchain, can be modified. Timestamps for every block are generated on each transfer; thus changing them is difficult. Each block has a parent block where the original information is stored. Figure 3 illustrates the block. The demand and usage of applications constructed on blockchain architecture is only expected to evolve.

2.2.2 Smart Contract

Nick Szabo, who presented the concept of cryptography in 1994, proposed the idea of creating smart contracts. The concept of smart contracts has quickly gained popularity in both academic and professional circles. The potential for smart contracts to radically change a number of industries, including the finance and supply chain management, is one key theme. Researchers have investigated both the legal and regulatory implications of using smart contracts as well as their technical aspects, such as the programming languages and execution protocols. Studies have also concentrated on the security flaws of smart contracts and suggested remedies to reduce the risks. Overall, the research indicates that smart contracts have a significant potential to increase efficiency and transparency in many different fields. According to the given definition of a contract, it is a legitimate arrangement between the gatherings that portrays their common obligations. Smart contracts can be built on various blockchain platforms, including *Ethereum*, *EOS*, and *Hyperledger Fabric*. Each platform has its own programming language, such as *Solidity* for *Ethereum*, *C++* for *EOS*, and *Go* for *Hyperledger Fabric* [28]. Smart contracts are self-executing contracts which contain the terms of an agreement between two parties, written in code. These contracts are stored on a blockchain network and are executed automatically when certain conditions have been met. The purpose of smart contracts is to automate the process of executing agreements, eliminating the need for intermediaries, and providing greater security and transparency. Smart contracts have the potential to revolutionize the way agreements are executed, thus providing a greater efficiency, security, and transparency [29]. **Table 2.** presents the properties of smart contracts for different platforms.

Table 2. Smart contracts comparison with multiple platforms

Property	Ethereum	Hyperledger Fabric	EOS
Language	Solidity, Vyper	Chaincode (Go, Java)	C++, WebAssembly
Consensus Mechanism	PoW, transitioning to PoS	Pluggable (multiple options)	DPoS

Permission Model	Public, permissionless	Private, permissioned	Public with configurable permission levels
Scalability	Scaling challenges	Modular architecture for scalability	High throughput (but less decentralized)
Flexibility	Wide range of use cases	Enterprise-focused with configurable features	High-performance decentralized applications
Governance	Decentralized governance (EIPs)	Consensus-based governance	EOS.IO Core Arbitration Forum
Interoperability	Interoperable with other blockchains	Supports interoperability with other frameworks	Interoperability through inter-blockchain communication
Development Community	Large and active	Active enterprise-focused	Growing community
Use Cases	Diverse range of applications	Enterprise-grade applications	Decentralized applications and exchanges

2.2.3 Zero-Knowledge proof

Zero-knowledge proof is a cryptographic concept which enables a prover to prove the truth of a statement to a verifier without revealing any additional information. In other words, zero-knowledge proof allows for the authentication of a statement without revealing the underlying data, thereby ensuring privacy. There are various kinds of zero-knowledge proofs, including interactive and non-interactive proofs, as well as proof of knowledge proofs. Interactive zero-knowledge proofs need communication between the prover and the verifier, whereas non-interactive zero-knowledge proofs do not require that. Zero-knowledge proof has various applications, including blockchain technology, where it is used to enhance the privacy of transactions. In blockchain, a zero-knowledge proof can be used to prove the ownership of a certain asset without revealing the identity of the owner [30]. **Table 3.** illustrates a brief description of use cases.

Table 3. Zero-knowledge proof descriptive analysis

Use Case	Description
Authentication [31]	Zero-knowledge proof can be used to authenticate a user's identity without the need for a password or other identifying information.
Secure Data Transfer [32]	Zero-knowledge proof can be used to transfer-sensitive data securely without the need for a trusted intermediary.
Anonymous Transactions [33]	Zero-knowledge proof can be used to allow anonymous transactions while still maintaining the integrity of the transaction ledger.
Digital Signatures [34]	Zero-knowledge proof can be used to create digital signatures that are both secure and unforgeable.
Password-less Login [35]	Zero-knowledge proof can be used to enable passwordless login to online services.

2.2.4 Consensus Mechanisms

Consensus mechanisms are used to resolve consistency issues associated with distributed systems [36]. A consensus-based protection algorithm is a technique used to ensure that a particular operation is protected for a limited time, consistent, acknowledged, and tamper-proof. In a blockchain network, a consensus mechanism is required to ensure communication among peer nodes as well as to ensure transaction validity. All requirements cannot be met by one type of consensus. A consensus on a blockchain can be categorized into two types – public and private. These days, users prefer to conduct simple surveys or elections online. An election could be tampered by a cyberattack or a hack. This technology provides a means to safeguard the privacy of users by protecting their identity and allowing for secure record keeping. A public blockchain example of a consensus mechanism uses *Proof of Work* (PoW) [37], *Proof of Stake* (PoS) [38], *Proof of Authority* (PoA) [39] and *Proof of Identity* (PoI) [40]. Regarding the significance of putting in place a secure method to prevent modifications or transactions in the blockchain network to maintain the integrity and stability of the blockchain, this method should need the approval of all the participating nodes [41].

2.3 Federated Learning (FL)

Federated learning is a distributed machine learning approach which allows models to be trained on decentralized data without requiring data exchange. Mixing machine learning with federated learning can provide various advantages, including higher privacy, lower communication costs, and improved scalability. This combination is possible by training machine learning models on data that is

strategically spread across several devices using federated learning methods. The trained models are then combined and updated to provide a final model that is more accurate than the model of any single device. The concept of federated learning is not new as Patrick Hill, a philosophy professor, developed the *Federated Learning Community* (FLC) in 1976 to help students overcome anonymity and isolation [42]. In *Federated Learning* (FL), technology is used to overcome the challenges of data accessibility through decentralized segregation. Data has grown increasingly important to every firm in recent years, and ensuring data security and privacy is a major problem [43]. A classic machine learning technique entails training algorithms on a central server with all accessible training data. Nevertheless, it has a number of drawbacks: consumers demand immediate responses, yet the connectivity between a device and a central server might be delayed. Edge computing makes continuous learning more difficult. Personal information may get compromised.

The Federated Learning model for online payments secures the identity and personal information. In Lithuania, the *Data Protection Act* of 2018 is in effect to safeguard user data. The primary legislative document directly governing and establishing the general requirements for data protection in Lithuania is the *General Data Protection Regulation* (GDPR), also known as Regulation (EU) 2016/679 [44]. By adopting federated learning, models may be trained without delaying the process. Federated learning can advance research on artificial intelligence. Users can create personalized models using federated learning without having their privacy invaded. *Google* put up a data federated modeling plan for *Android* phone model updates in 2016 [45]. Federated learning is a powerful tool for enabling collaborative machine learning while preserving data privacy and security. Yet, it is prominently distinct from the conventional techniques for large data privacy protection, such as differential privacy and anonymity. By communicating encrypted processing parameters, a federated learning (FL) system preserves user privacy and prevents hackers from accessing private information. The General Data Protection Regulation (GDPR) and other regulations have not been broken as long as the data used in federated learning is kept secure and is not identifiable. Additionally, users must be informed about the purpose and risks of data processing. Federated learning can be a safe and effective way to build machine learning models while protecting user privacy.

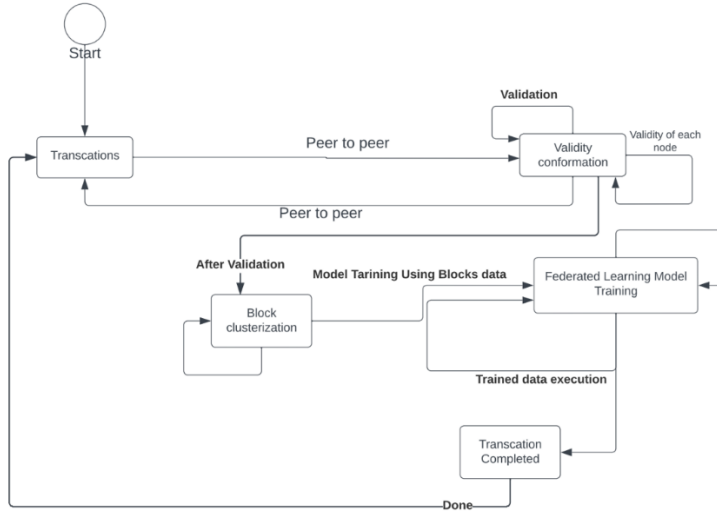


Fig. 4. Transaction process using Federated Learning

Transaction process using federated learning is explained in Fig. 4. as it starts transactions and leads to peer to peer. The initial step is to choose a model that has either been pre-trained, or has never been trained. The basic model is then delivered to a local device or a local server. Local machine learning models are trained by using local datasets. It is essential to create a global model. Global models employ aggregate information to identify the most effective performance. The global model features are communicated with the local data centers in order to incorporate global models into the local model by using federated learning in the transaction process. This approach ensures that the models are constantly updated and optimized, which leads to increased performance and accuracy over time. Third-party intermediaries are not needed because of their decentralized architecture. Validity confirmation for transactions in public blockchain transactions are added by consensus. After validity confirmation, the clustering method is used for deanonymizing the blockchain data.

2.3.1 Federating Learning and Blockchain

Blockchain can endure single point failures and distributed consensus enabling the use of incentive mechanisms that successfully boost participation in the system. A blockchain based FL architecture was developed to verify the uploaded parameters and investigate related system performance issues. Blockchain features, such as tamper-proofing, anonymity, and traceability, can be used to audit machine learning (ML) models for increased trustworthiness. Decentralized FL frameworks are being used to investigate security and privacy issues. Blockchain is being used to solve conventional FL problems. A blockchain can be deployed in FL to realize decentralization, so the central

aggregator can be replaced by a *peer-to-peer* (P2P) blockchain system, so the blockchain nodes can aggregate global models. Consequently, there is no unreliability of the whole FL system when the server fails [46]. Federated learning (FL) is a distributed machine learning technique which allows several devices to train a model cooperatively without requiring raw data to be shared with a central server. FL architecture generally has three important components as client devices, which are the devices used in the training process and can include smartphones, laptops, or other electronic devices. The second component is the edge, which is responsible for processing data and making decisions in real-time. The final component is the cloud, which stores and manages the data collected from the client devices and the edge. Also, blockchains are being used to solve conventional FL problems. Essentially, a blockchain is a distributed decentralized database. In the system, all interactive records (transactions) are linked into chains as blocks and stored in each section according to their time sequence [47]. A blockchain can be deployed in FL to realize decentralization, so the central aggregator can be replaced by a peer-to-peer blockchain system. Thus the blockchain nodes can aggregate global models, and therefore there is no unreliability of the whole FL system when the server fails. **Table 4.** provides a comparison between Federated Learning and Blockchain

Table 4: Comparison table between Federated Learning and Blockchain

Criteria	Federated Learning	Blockchain
Purpose	Collaborative machine learning	Distributed ledger technology
Data Privacy	Private data remains on devices	Publicly accessible data
Scalability	Can handle large datasets	Scalable for small to medium datasets
Consensus Mechanism	No consensus mechanism	Requires consensus mechanism for data validation
Security	Strong security guarantees	Security depends on the consensus mechanism and implementation
Use Cases	Healthcare, Finance, Transportation	Cryptocurrency, Smart Contracts, Supply Chain Management

Blockchain and Federated Learning can be combined to create a secure and decentralized approach to machine learning. In this approach, the blockchain acts as a secure ledger which records and verifies transactions between the participants of the federated learning process. The Custom Blockchain’s enhanced programmability, extensibility, and flexibility are the primary drivers. In particular, BCFL offers an architecture built on a custom blockchain in which local learning

model changes are shared and verified [48]. The system is made up of a dual module which combines a FL module with a permission blockchain module, as presented in **Fig. 5.** [49]. The blockchain network, the federated learning network, and the machine learning algorithm make up the three key parts of the architecture of blockchain-based federated learning. Many users in the federated learning network add their local data to the machine learning challenge. Data gathering from the present process, which enables cross-verification, block propagation, and block production, is shown below. The participants create model changes by using their local data and broadcast them to the blockchain network. The participants also get model updates from the blockchain, which they then apply to their regional models to make them better.

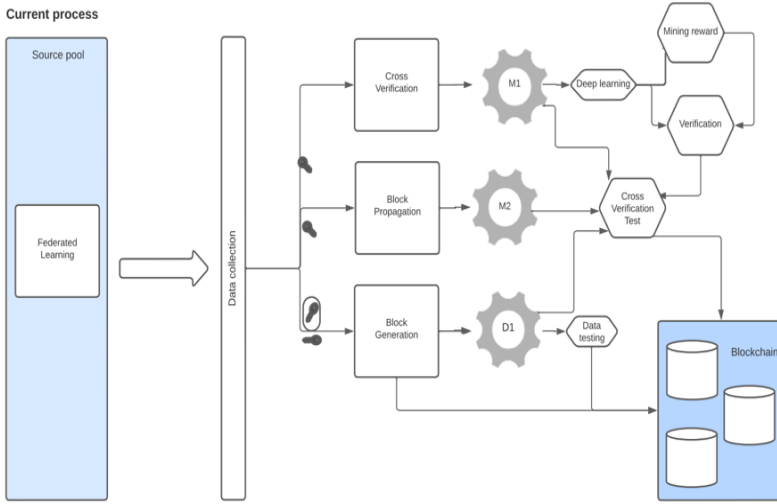


Fig. 5. Architecture of Blockchain Based Federated Learning

In essence, FL is a form of machine learning. As a result, its effectiveness, efficiency, and security are crucial factors to consider. Due to this, several studies have been suggested to strengthen BCFL and the performance of the model. FL is a distributed machine learning technique which allows local data storage. Through interactive gradient values, the client implements training in this approach. As a result, the fundamental concept behind increasing the model's accuracy is comparable to that of traditional machine learning.

2.3.2 Industrial Applications of Blockchain and Federated Learning

Federated Learning is a machine learning technique which allows multiple devices to collaborate on a common machine learning model.

This can be particularly useful in industrial applications where data is generated by

multiple devices or sensors. One of the most significant advantages of Federated Learning is that it can help to overcome the data silo problem. In many industrial applications, data is generated by multiple devices or sensors, each with their own unique data format. This can make it challenging to create a unified machine learning model. Federated Learning can help to overcome this problem by allowing multiple devices to collaborate on a common model. Cybersecurity is also important to FL; it preserves the data stored on the device. The model is only shared across connected networks when it is updated. Qu *et al.* [50] discussed fog computing and IoT problems that could be solved with blockchains. It was found that fog computing works as a combination of cloud computing and edge computing, and that it could reduce network congestion problems. Lu *et al.* [51] also proposed a federate learning architecture for providers to reduce the privacy concern associated with addresses and the volume of transmissions.

Table 5. Blockchain and Federated Learning Implementation in industrial sectors

Industry	Blockchain Implementation	Federated Learning Implementation
Supply Chain Management	Tracking products and materials through the supply chain, reducing fraud and increasing transparency.	Collaborative model training with multiple parties without sharing data.
Healthcare	Securing medical records and patient data, improving data management and privacy.	Collaborative model training with multiple hospitals and clinics without sharing sensitive patient data.
Finance	Improving security and transparency in financial transactions, reducing fraud.	Collaborative model training with multiple financial institutions without sharing sensitive financial data.
Manufacturing	Tracking the movement and quality of goods and materials, optimizing supply chain efficiency.	Collaborative model training with multiple manufacturers to improve product quality and reduce defects.
Energy	Improving the efficiency of energy distribution and management, reducing waste and costs.	Collaborative model training with multiple energy companies to optimize energy production and distribution.

A componential model was also developed for improving the reliability and security of model parameters, which included DAGs and permitted Blockchains [52]. Furthermore, a framework for enhancing the efficiency of Deep Reinforcement Learning was proposed by Zhou *et al.* [53] also proposed BlockFL architecture for solving the problem of data security for those users who have had their personal information collected by companies. By using blockchain-based learning, they

proposed a system for training smart driving models without sharing data. By integrating FL and blockchain, we can solve resource consumption limitations in centralized systems, as well as in sensitive autonomous vehicle applications. In comparison to centralized networks, the decentralized solution will optimize computing resource utilization [54]. In addition to strengthening trust and providing incentive mechanisms that reward good behavior and detecting malicious users, a blockchain-powered FL system can overcome such challenges as poisoning attacks, inference attacks, and backdoor attacks [55].

In **Table 5.**, it is outlined that there is the potential to revolutionize data storage, privacy, and security in various industrial applications for a number of Industry Sectors with a combination of blockchain and Federated Learning. By implementing these technologies, we can create a more secure and decentralized approach to machine learning which can improve efficiency, reduce costs, and enhance privacy and security.

2.4 Anomaly Detection

In today's data-driven world, businesses are generating massive amounts of data on a daily basis. This data is often used to make informed decisions, optimize processes, and gain valuable insights. However, with the increasing volume and complexity of the data, it can be challenging to identify patterns and outliers that indicate an anomaly in the system. This is where anomaly detection comes in. Anomaly detection is a technique used to identify unusual patterns or those events which deviate from the norm. It involves analyzing data to look for patterns and trends that do not conform to the expected behavior. Anomalies can be caused by a variety of factors, such as errors in data collection, fraud, cyber-attacks, or system failures. Attacks and anomalies are common occurrences in cyberspace, and they can impact practically everything mechanically or electronically connected or requiring the networking to function. To detect mistakes like this, a dynamic system which detects the attack from the start, before it reaches the host or client domain, is required. Since the development of deep learning, there has been a greater emphasis on neural networks which are required to learn sensor data that reflects a latent *Internet of Things* (IoT) operational network. System logging is used to find anomalies and deviations from the expected execution patterns. This paradigm is based on the system logging, which enables the identification of anomalies and departures from the conventional patterns of execution. System logging is necessary for this paradigm since it allows the learner to analyze the pattern, and also allows the system to recognize the pattern, the determinants of the threshold, and to find out what categorizes the anomaly, in accordance with Lane *et al.* [56].

The anomaly classifier will provide the inference and forecast what the anomalous data point that has failed to pass through the threshold determinant which is based on the value of the threshold. The anomaly classifier will provide the inference and anticipate what the aberrant data point is that did not pass through the

threshold determinant once it has been identified, subject to the anomaly detection system's conscription threshold [57]. Due to their ability to comprehend the relationships between networks, which are always temporal, long-short-term memory networks in federated learning are specifically configured to detect anomalies. A general overview of the foundations of spoofing on the internet was provided by Rashid *et al.* [58]. The data that is conveyed over the internet is first divided into a number of packets which are then communicated independently and reassembled. Each bundle contains an IP header that contains details about the bundle, including the source and destination IP addresses. In IP mocking, the source address is changed in the bundle header to deceive the computer system into accepting the package as coming from a reliable source, such as another computer on a legitimate network.

2.4.1 Traditional ways of anomaly detection

Analysis of data sets with the objective to find patterns and anomalies is a traditional step in anomaly detection. This often involves statistical analysis to find patterns in the data and determine the likelihood that specific occurrences will occur. In order to detect abnormalities in real-time, machine learning algorithms can also be trained on previous data. The statistical technique is one of the conventional approaches of anomaly detection that is most frequently utilized. With this approach, data sets are analyzed to ascertain the likelihood that specific occurrences will transpire. An occurrence could be labeled as an anomaly, for instance, if it happens far more frequently than is normal. The machine learning method is an alternative strategy that uses past data to train computers to identify abnormalities in real-time. For working with big data sets, this strategy is quite effective. According to a survey by Ahmed and Hu [59], it has been revealed that there are many methods for finding network anomalies, including statistical, clustering, classification and information theory-based methods. In this literature review, the common approaches based on categorization shall be described first. The construction of a knowledge base containing historical information on both malicious and benign records is the fundamental aim of classification-based anomaly detection. A comparable attack against the system will be promptly discovered. The disadvantage of the classification-based approach is that an attack on anomaly detection only succeeds if the model has previously categorized it. The importance of classification-based approaches in anomaly detection involves constructing a knowledge base of historical data on both malicious and benign records. However, the disadvantage of this approach is that it fails to detect new threats that have not been previously categorized. Therefore, significant training and effort are required to build a robust knowledge base that can effectively detect anomalies.

The traditional ways of anomaly detection are Statistical Analysis, Machine Learning Algorithms, Rule-based Systems, Clustering Techniques, Neural Networks, Neural Networks, Decision Trees and Support Vector Machines. These techniques entail looking for patterns and abnormalities in historical data. They have

limits, but still they have been useful to some extent. For instance, statistical analysis makes the assumption that the distribution of the data is normal, which is not necessarily the case. For machine learning algorithms to work well, a large amount of labeled data is needed. Rule-based systems might not be able to detect all kinds of abnormalities, and they can be rigid. Neural networks are susceptible to overfitting, and clustering procedures can be computationally costly. Complex interactions between variables might not be captured by decision trees and support vector machines. A paper by Eskin *et al.* [60] was aimed to investigate the feature space of the data, for which an unsupervised model was used that could identify the anomalous events so that to measure the feature space and determine if the data is benign. Thus the model can identify anomalous events. However, the findings demonstrate that not all attacks can be precisely recognized. This is due to the possibility that abnormal data coexists with benign data in the same feature space. Unsupervised learning has the drawback that the existence of a pure benign dataset cannot be ensured. There is a supervised approach which enhances the detection performance and is comparable to the One-Class SVM and classification-based techniques.

2.4.2 Supervised Learning Approach

A labeled dataset is used to train the algorithm in the machine learning technique known as supervised learning. This indicates that the dataset contains input characteristics and the associated output labels that the algorithm learns to anticipate [61]. Creating a model that can precisely predict the output label for fresh input data is the aim of supervised learning. Splitting the labeled dataset into the training and testing sets is a key step in the supervised learning process. The algorithm is trained by using the training set, and the model's performance is assessed by using the testing set. The algorithm modifies its parameters in response to the input characteristics and the output labels to learn from the training set. Compared to other machine learning techniques, supervised learning provides a number of benefits. It is simple to build and has a variety of uses, such as speech and picture recognition, natural language processing, and recommendation systems. Moreover, it makes the unavoidable assumption that the input data is independent and uniformly distributed.

In **Table 6.**, we presented methods of Supervised learning that can be used for anomaly detection.

Table 6. Methods of Supervised learning for anomaly detection

Method	Description	Advantages	Limitation
Classification	Labels data as normal or anomalous based on a predefined threshold.	Easy to implement, can handle large datasets.	Assumes that data is independent and identically distributed; threshold selection

			can be difficult.
Regression	Models the relationship between input variables and their expected output, detects anomalies based on deviations from the expected output.	Can handle complex relationships between variables, can be used for both continuous and categorical data.	Assumes linear relationship between the input variables and the output, may require significant feature engineering.
One-Class Classification	Learns a model of the normal data and identifies any data points which deviate from that model as anomalous.	Does not require labeled anomalous data, can handle complex data distributions.	Does not generalize well to new data, can be sensitive to outliers in the normal data.

Anomaly detection methods based on supervised learning have their own drawbacks, such as the requirement for huge quantities of labeled data and the presumption of independently and uniformly distributed input data. Supervised learning could be a potent tool for finding abnormalities in a range of applications when utilized.

2.4.3 *Unsupervised Learning Approach*

Unsupervised learning is a sort of machine learning in which an algorithm is trained on an unlabeled dataset. Unsupervised learning algorithms attempt to find patterns and correlations within data without any prior understanding of what they represent, as opposed to supervised learning algorithms which label the data and train the algorithm to predict the output for fresh inputs [62]. Clustering, dimensionality reduction, and anomaly detection are examples of unsupervised learning methods. Clustering methods gather comparable data points based on their qualities, whereas dimensionality reduction algorithms minimize the data complexity by selecting the most essential aspects. Anomaly detection techniques discover outliers in the data in the sense that they do not fit the anticipated patterns.

There are various advantages to unsupervised learning versus supervised learning. When no labeled data is available, it can identify previously unrecognized patterns and correlations in the data. Unsupervised learning can also be used to preprocess data before using supervised learning algorithms, which can enhance their performance. Unsupervised learning provides a powerful tool for analyzing large

and complex datasets and can be particularly useful in situations where labeled data is scarce or difficult to obtain [63]. By identifying hidden patterns and relationships in the data, unsupervised learning can help researchers and businesses make more informed decisions and gain new insights into their data. **Table 7.** illustrates the advantages and limitations of the unsupervised learning approach.

Table 7. Methods of Unsupervised learning for anomaly detection

Unsupervised Learning Approach	Advantages	Limitation
Clustering	Can identify groups of data points without prior knowledge of what those groups should be.	Not able to identify anomalies that do not fit neatly into any cluster.
Density-based methods	Can detect anomalies in regions of low density.	Not able to detect anomalies in regions of high density.
Principal Component Analysis (PCA)	Can identify linear combinations of variables that explain the most variance in the data.	Assumes that anomalies are outliers in the low-dimensional space defined by the principal components.
Autoencoders	Can learn a compressed representation of the data that captures its most important features.	Not able to detect anomalies that are not well-represented by the learned representation.
Isolation Forests	Can isolate anomalies by randomly partitioning the data.	Can be computationally expensive for large datasets.

2.4.4 *Support Vector Machine*

Support Vector Machine (SVM) is a strong machine learning technique that is used for classification and regression analysis [64]. It is a supervised learning method that is often used to solve complicated classification issues. SVM works by determining the best feasible boundary (hyperplane) between data points from distinct classes. The hyperplane is set in such a way that the margin between the classes is maximized. SVM is widely used to solve classification issues in a variety of disciplines, including image classification, text classification, and bioinformatics [65]. SVM is special in that it can efficiently handle high-dimensional data. SVM is also resistant to data noise and anomalies. Another benefit is that SVM offers a one-of-a-kind solution and is less prone to overfitting than other machine learning algorithms. SVM also deals with non-linearly separable data by employing kernel functions, such as polynomials, radial basis functions (RBF), and sigmoid [66]. SVM, on the other hand, suffers from several restrictions. One of the disadvantages

of SVM is that its performance is strongly dependent on the kernel function and its parameters. Choosing the incorrect kernel function or settings might result in poor performance. SVM is also computationally demanding, as it necessitates a large amount of memory to store the training data. Another issue is that SVM is not ideal for handling huge datasets since training the model takes a long time.

2.4.5 *Clustering-based Methods*

Clustering-based anomaly detection methods, such as K-means, have gained popularity in the field of anomaly detection due to their ability to identify outliers based on the data's clustering structure. Numerous research publications have investigated the use of clustering algorithms for anomaly detection, thereby making major contributions to the field. This is a framework which combines K-means clustering with density estimation to find abnormalities in high-dimensional data. This method outperforms others in detecting anomalies in a variety of domains, including network intrusion detection and credit card fraud detection [67]. The K-means technique uses similarity measurements to divide data into multiple clusters, and anomalies are frequently discovered as data points that do not belong to any cluster or stray significantly from the cluster centroids. To improve the accuracy of anomaly identification, researchers have extended the classic K-means by including such techniques as distance-based outlier ratings or density estimation. Metrics, such as purity, entropy, and precision at top-n, are used to assess the performance of clustering-based anomaly detection systems. Clustering-based anomaly detection algorithms, particularly K-means, offer a beneficial way for detecting abnormalities based on the underlying data structure and remain an active study area in the field of anomaly detection. Anomaly detection approaches known as 'clustering-based methods' group data points based on their similarity [68]. Anomalies are data points that do not belong to any cluster or that belong to a cluster with an abnormally small number of data points relative to the other clusters. A popular clustering-based technique, K-means clustering, separates data points into k groups depending on their distance from the cluster centroids [69]. Anomalies are data points that are far from any centroid or do not fit into any cluster. Hierarchical clustering is another clustering-based technique which arranges nested clusters into a tree-like structure based on how related they are. The data points that are not part of a cluster or that make up a small, isolated cluster can be used to detect anomalies.

2.4.6 **Autoencoder**

In recent years, autoencoder-based anomaly detection has emerged as a notable technique, gaining significant academic interest. Autoencoders, a sort of neural network design, have shown promise in capturing complicated data linkages and accurately spotting deviations from the regular patterns. In the domain of anomaly detection, many autoencoder architectures such as feedforward, convolutional, and recurrent, have been investigated. To improve the anomaly detection capabilities of autoencoders, researchers have studied various training strategies, such as

minimizing mean squared error or binary cross-entropy, and combining regularization techniques, such as sparsity constraints and denoising. Several research articles have contributed to this subject, including Huang *et al.* [70], which developed a deep autoencoder architecture with stacked sparse autoencoders and a SoftMax classifier attaining superior results on several datasets. Autoencoder-based anomaly detection systems are evaluated by using such measures as accuracy, precision, recall, F1 score, and AUC-ROC, as well as domain-specific assessment methodologies adapted to certain circumstances. Liu *et al.* [71] provided an evaluation methodology which integrates various metrics to provide a holistic assessment. Autoencoders provide such advantages as the capacity to detect global and local anomalies and capture non-linear correlations. As they do not rely on labeled data, they are particularly well suited for unsupervised anomaly detection scenarios. However, difficulties remain, including recognizing unexpected anomalies that differ significantly from the training data distribution. For the sake of addressing these limitations, investigation of hybrid approaches that combine autoencoders with other anomaly detection methods, and investigation of advanced training methodologies, such as adversarial training and reinforcement learning to improve the resilience and accuracy of autoencoder-based anomaly detection systems, could be the future research directions.

2.4.7 Recurrent neural network

For a number of reasons, including its ability to grasp temporal relationships and sequential patterns within data, recurrent neural networks (RNNs) have received a lot of interest in the field of anomaly identification. RNNs excel in detecting anomalies in time series data, such as sensor readings, network traffic, and financial transactions. Several research publications have investigated the use of RNNs for anomaly detection, thereby contributing to the advancement of this field. Wei *et al.* [72] suggested a technique which uses an RNN-based encoder-decoder architecture to recreate the typical patterns and identify abnormalities based on reconstruction error. Their research showed encouraging results in spotting anomalies in a variety of fields, including healthcare and cybersecurity. The architecture of the RNN, such as *Long Short-Term Memory* (LSTM) or *Gated Recurrent Unit* (GRU), is determined by the unique needs of the anomaly detection task [73]. The researchers have also looked into hybrid architectures that mix RNNs with additional neural network components, such as convolutional layers to capture both temporal and spatial information in data. Metrics like accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC) are used to analyze the effectiveness of RNN-based anomaly detection systems [74]. Furthermore, such techniques as sliding windows and ensemble methods have been used to deal with long-term dependencies and improve the RNN detection accuracy. Despite their efficacy, RNNs suffer difficulties due to vanishing or exploding gradients, which can impair the training stability and model performance.

2.4.8 Anomaly Detection with Federated Learning and Blockchain

Anomaly detection is a fundamental problem in machine learning that includes spotting unusual occurrences or trends. Federated learning and blockchain are two new technologies that can increase the accuracy and security of anomaly detection systems. Federated learning is a machine learning technique that allows several devices to train a model cooperatively without sharing sensitive data with a central server. This method is especially beneficial for detecting anomalies where data privacy is an issue. Federated learning can lessen the risk of data breaches and safeguard user privacy by training models directly on user devices. In contrast, blockchain is a decentralized ledger that is used to safeguard and verify transactions. By incorporating federated learning and blockchain technologies, anomaly detection systems can become more accurate and secure. Federated learning allows for collaborative training without compromising data privacy, while blockchain provides a decentralized and secure way to verify transactions [75]. Together, these technologies can enhance the effectiveness of anomaly detection and improve the overall data security. Information security has increased the significance of the difficulty of training on decentralized data. Federated learning makes it possible for each device to conduct training in secret and report the findings to the main server, while still maintaining anonymity. By using federated learning to train models on local devices and blockchain to secure and verify data and model updates, the system is able to achieve high accuracy while maintaining data privacy and auditability. Sater and Hamza [76] have published federated learning research on anomaly detection. The proposed approach outperforms the traditional centralized anomaly detection approaches by reducing the communication overhead and preserving data privacy. The federated learning technique is excellent for demonstrations on simulated datasets, due to allowing for equitable distribution of real data from a General Electric smart building. The model in the current work is trained similarly using federated averaging, and iterations are carried out until the model converges or the maximum number of training rounds is reached. The training approach comprises entering data from various heterogeneous data sources across several sensors to solve the common heterogeneous concerns of IoT devices.

Lu *et al.* [77] created a distributed and immutable audit trail. For the aggregate, the study also uses the FedAvg algorithm. Based on the performance of the deep autoencoder, this work shall use the deep autoencoder model as the local model and apply a similar configuration for the loss function, metrics, and data preparation. The audit frequency of global model updates and the local rate of convergence on clients are discovered to be in trade-off. According to the findings cited, the federated learning environment continues to encounter a number of challenges.

Yeganeh *et al.* [78] suggested several techniques for dealing with non-IID data, such as data augmentation, creating a compact dataset that can be shared worldwide, changing the ability to complete training on local data on each device, and putting an

assumption on data dissimilarities. Sheller *et al.* [79] created an unbalanced dataset to imitate a genuine dataset in order to compare the local model with the FL model and assess if the FL model improves the performance in the absence of sufficient data. Furthermore, it ensures that cybersecurity businesses can create a general model from a small number of data points in real-world applications, which is a significant improvement for blockchain intrusion detection. Anomaly detection might serve as a bridge to discovering strategies to counter such attacks. According to this situation, the FL system only accepts the test mistake if it enhances the overall model, and rejects it otherwise. These proactive steps activate the FL system's defensive systems so that to identify erroneous model updates and mitigate their effects.

It has been proposed that if the client or host is compromised, the attacker can entirely manipulate the training process consisting of data, algorithms, and parameters, which are constraint and scale attack [80]. If there is a change in the training data label setup for the system, the attacker's goal of anticipating a change surrounding the decision limits or having the notion of building up the network drift is achieved by influencing the model of the learning inputs. The developer can build synthetic guiding inputs that are designed to slow down or limit the capacity by adding information to samples before extracting the nearby feature or developing a framework to cause a gradual drift in the notion. To defend against such an invasion, the developer must employ the most effective training techniques imaginable. Furthermore, as compared to the non-FL model, our recommended approach improves the reliability and efficiency of blockchain anomaly detection. Individuals deployed on it [81] presented some potential mitigation strategies to develop a new federated learning defense for poisoning attacks. This could be achieved by looking for an improved feature and using a clustering approach to spot the model updates that are corrupted. Security gateways are expected to have total control over the data used for training, introduce filtered material that is toxic, and occasionally permit an approach; thus, all of these are feasible solutions and are required. The major goal of the defensive ideas is to enable the users to identify incoming updates that are hostile, suspicious, and which originate from questionable sources. A model update that has been separated into two (2) groups based on the features described is one example of this; the smaller group is identified as the harmful one and is then filtered out. A defensive and spoofing approach was presented by Guo *et al.* FF, in which there is a very brief conversation between the client and the host. In order to find anomalous data and complete the desired end state, the federated learning technique has been applied to the model. By comparing the two approaches, Federated learning has the advantage of preserving data privacy by keeping the data locally, while blockchain ensures data integrity and transparency. Federated learning can be more efficient than blockchain as it requires less computational power to maintain the model.

Table 8. Anomaly detection by using different techniques

Reference	Key Features	Techniques Used	Dataset	Performance
[82]	Federated anomaly detection system for IoT networks	Blockchain, zero-knowledge proof	Real-world IoT datasets	Accurate anomaly detection using KNN and SVM
[83]	Blockchain-enabled federated learning with secure aggregation	Blockchain, differential privacy	Synthetic dataset	Improved anomaly detection with KNN and SVM
[84]	Federated anomaly detection system with privacy-preserving	Homomorphic encryption, differential privacy	Real-world datasets	Effective anomaly detection using autoencoder and RNN
[85]	Zero-knowledge proof-based mechanism for knowledge transfer	Zero-knowledge proof	Synthetic dataset	Secure knowledge transfer using autoencoder and RNN
[86]	Federated learning framework with blockchain-preserved privacy	Blockchain	Real-world dataset	Accurate anomaly detection with KNN and SVM
[87]	Privacy-preserving anomaly detection in federated learning	Secure multi-party computation, differential privacy	Synthetic dataset	Enhanced privacy protection with autoencoder and RNN
[88]	Blockchain-based anomaly detection for IoT devices	Blockchain, anomaly detection algorithms	Real-world IoT datasets	Improved anomaly detection accuracy and efficiency
[89]	Privacy-preserving federated learning for anomaly detection	Differential privacy, federated learning	Synthetic dataset	Preserved privacy and accurate anomaly detection
[90]	Federated anomaly detection with distributed data	Collaborative learning, data partitioning	Real-world datasets	Efficient anomaly detection across distributed data sources
[91]	Anomaly detection in federated sensor networks	Sensor fusion, anomaly detection algorithms	Real-world sensor datasets	Enhanced anomaly detection in federated environments

[92]	Secure anomaly detection in federated wireless sensor networks	Encryption, distributed anomaly detection	Real-world sensor datasets	Privacy-preserving anomaly detection in sensor networks
[93]	Federated learning-based anomaly detection in cloud computing	Federated learning, cloud computing	Synthetic dataset	Improved anomaly detection in cloud computing environments
[94]	Privacy-aware anomaly detection in federated wireless networks	Privacy-preserving techniques, anomaly detection algorithms	Synthetic dataset	Preserved privacy and accurate anomaly detection
[95]	Federated anomaly detection with trust-based aggregation	Trust management, aggregation algorithms	Real-world datasets	Reliable anomaly detection with trust-based mechanisms
[96]	Distributed anomaly detection using federated learning	Federated learning, distributed data analysis	Synthetic dataset	Efficient distributed anomaly detection
[97]	Privacy-preserving anomaly detection in federated databases	Privacy-preserving techniques, anomaly detection algorithms	Synthetic dataset	Enhanced security for privacy that is accurate
[98]	Federated anomaly detection using K-means clustering	K-means clustering	Real-world datasets	Effective anomaly detection with K-means clustering
[99]	Privacy-preserving anomaly detection in federated databases	Privacy-preserving techniques, anomaly detection algorithms	Synthetic dataset	Enhanced privacy protection with accurate anomaly detection
[100]	Federated learning-based intrusion detection in ad hoc networks	Federated learning, intrusion detection	Synthetic dataset	Improved intrusion detection accuracy in ad hoc networks
[101]	Secure anomaly detection in federated sensor networks	Secure aggregation, anomaly detection	Real-world sensor datasets	Privacy-preserving anomaly detection in sensor networks

		algorithms		
[102]	Federated learning framework for distributed anomaly detection	Collaborative learning, distributed data analysis	Real-world datasets	Efficient distributed anomaly detection across nodes
[103]	Privacy-aware anomaly detection in federated sensor networks	Privacy-preserving techniques, anomaly detection algorithms	Synthetic dataset	Preserved privacy and accurate anomaly detection
[104]	Federated anomaly detection with distributed data fusion	Data fusion, distributed anomaly detection	Real-world datasets	Reliable anomaly detection with distributed data fusion
[105]	Federated learning-based intrusion detection in cloud computing	Federated learning, cloud computing	Synthetic dataset	Improved intrusion detection accuracy in cloud computing
[106]	Privacy-preserving anomaly detection in federated wireless networks	Privacy-preserving techniques, anomaly detection algorithms	Synthetic dataset	Preserved privacy and accurate anomaly detection
[107]	Federated anomaly detection with trust-based aggregation	Trust management, aggregation algorithms	Real-world datasets	Reliable anomaly detection with trust-based mechanisms
[108]	Distributed anomaly detection using federated learning	Federated learning, distributed data analysis	Synthetic dataset	Efficient distributed anomaly detection
[109]	Privacy-preserving anomaly detection in federated databases	Privacy-preserving techniques, anomaly detection algorithms	Synthetic dataset	Enhanced privacy protection with accurate anomaly detection
[110]	Federated anomaly detection using K-means clustering	K-means clustering	Real-world datasets	Effective anomaly detection with K-means clustering
[111]	Privacy-preserving anomaly detection in	Privacy-preserving	Synthetic dataset	Enhanced privacy protection with

	federated databases	techniques, anomaly detection algorithms		accurate anomaly detection
[112]	Federated learning- based intrusion detection in ad hoc networks	Federated learning, intrusion detection	Synthetic dataset	Improved intrusion detection accuracy in ad hoc networks
[113]	Secure anomaly detection in federated sensor networks	Secure aggregation, anomaly detection algorithms	Real- world sensor datasets	Privacy-preserving anomaly detection in sensor networks
[114]	Secure anomaly detection in federated sensor networks	Secure aggregation, anomaly detection algorithms	Real- world sensor datasets	Privacy-preserving anomaly detection in sensor networks

Table 8. summarizes research publications on anomaly detection. In these papers, researchers have investigated various elements of anomaly detection in federated learning and related topics. To address the challenges associated with anomaly detection, they employ a variety of techniques, including the use of blockchain for secure and decentralized data management, zero-knowledge proof for authentication and privacy preservation, secure aggregation for protected data sharing, and differential privacy for enhancing privacy guarantees. The datasets used in the studies of these academics range from real-world datasets to synthetic datasets created particularly for the purpose of the research. The use of real-world datasets assures that the offered approaches are applicable and relevant to practical circumstances, whilst synthetic datasets allow for controlled experiments and comparison analysis. A major factor highlighted in the table is the performance of the recommended approaches. Researchers assess their methods by using a variety of performance metrics, including accuracy, privacy preservation, efficiency, and enhanced detection. To produce accurate anomaly detection results, specific algorithms or models, such as K-nearest neighbor's (KNN) and Support Vector Machines (SVM), are frequently used. This table provides significant insights into the varied variety of approaches and methodologies used in the field of anomaly detection in federated learning, blockchain, and privacy-preserving environments. These research papers improve knowledge and provide useful suggestions for designing successful anomaly detection systems which prioritize data privacy, security, and accuracy. There is not a single publication that would incorporate all the stated technologies (such as federated learning, privacy-preserving approaches, anomaly detection algorithms, blockchain, and so on) in a single research. There is

not a single publication that would incorporate all of the stated technologies (such as federated learning, privacy-preserving approaches, anomaly detection algorithms, blockchain, and so on) in a single research.

2.5 Section Summary

In this section, we have extensively discussed the current state in the sense that researchers have explored different techniques and algorithms to improve its effectiveness. Zero Knowledge Proof (ZKP) computations, supervised and unsupervised learning algorithms, clustering methods, such as K-means, Support Vector Machines (SVM), and Autoencoders, have been extensively studied and implemented for anomaly detection. Zero Knowledge Proof computations ensure the confidentiality and integrity of sensitive data, thus preventing unauthorized access or leakage. Supervised learning algorithms, such as SVM, create decision boundaries to separate normal instances from the outliers, while unsupervised learning algorithms like Autoencoders learn the underlying patterns and structures of the data in order to identify anomalies. Clustering-based methods, such as K-means, group similar instances together and detect deviations from the expected clusters. Additionally, the integration of blockchain and federated learning (FL) has shown promise in enhancing anomaly detection systems. The Blockchain technology provides a decentralized and tamper-resistant platform ensuring data integrity and traceability. FL enables collaborative model training without compromising data privacy, along with leveraging the collective knowledge of distributed devices or data sources. The combination of these techniques and technologies has the potential to revolutionize anomaly detection, enhancing security, privacy, and reliability. Ongoing research focuses on addressing challenges and improving the accuracy, efficiency, and scalability of anomaly detection methods. By incorporating these advancements, organizations can develop robust anomaly detection systems that effectively detect and mitigate anomalies in various domains.

3. RESEARCH DESIGN AND METHODS

3.1 Overview

It is critical to ensure a system's security, privacy, and decentralization while working with data and training models. To accomplish this purpose, several technologies must be integrated. The following section explains the methods, the suggested framework, and the model structure. The data combining technique is demonstrated through rigorous data collection and a sophisticated data pre-processing model. Furthermore, the use of *Support Vector Machines* (SVM) and auto-encoders enables the inclusion of anomalous data. These advanced techniques aim to improve the federated learning model's performance by detecting abnormalities in datasets.

3.2 Data Collection

The availability of high-quality datasets is critical to machine learning model training and performance. Data collection is used to build highly accurate and robust machine learning models, particularly in the field of visual recognition. Five prominent datasets, MNIST [115], E-MNIST [116], Fashion-MNIST [117], MNIST-C [118] and K-MNIST [119] were collected and gathered to train our model, as these datasets provide a diverse range of images. Moreover, a combination of several datasets yields a large training set that improves the model's performance and generalization skills.

3.2.1 MNIST

Due to the visualization and interpretability advantages of MNIST, which provide the opportunity to understand the behavior and performance of various classification algorithms, it is an ideal dataset to train our model on. It was combined by 500 individual authors who contributed to the MNIST Special Database (SD) 1 through SD-3, which is a collection of handwritten digits. It serves as a benchmark for any computer vision or machine learning problem; therefore, it would be perfect to train a model on it first.

After the dataset had been combined, several anti-aliasing and normalization methods were used to improve the clarity of the photos. The resulting group of pictures was then resized to be 28x28 pixels, and it is now referred to as the MNIST dataset. The MNIST dataset is available for download via the Keras Library, where it can then be loaded into programs. This simple benchmark dataset serves to test our program's model and its workings, as shown in **Fig. 6**

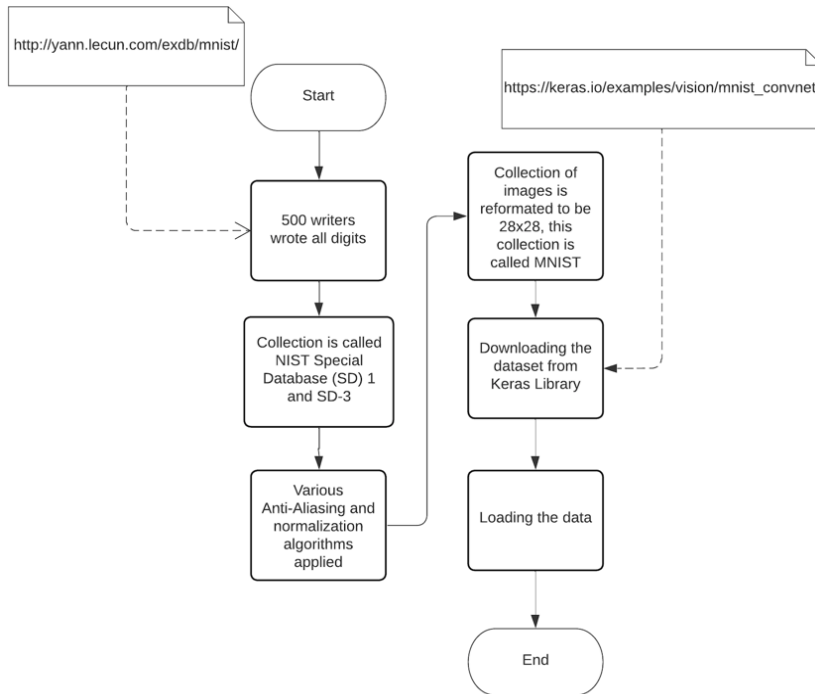


Fig. 6. Data Collection Visualization for MNIST dataset

3.2.2 E-MNIST

The entire English alphabet and the set of numerals are included in the enormous collection of handwritten characters known as the *NIST Special Database 19*. This database is special since it contains examples authored by a whopping 3600 authors, thereby creating a vast and varied collection of examples. To guarantee ideal clarity and consistency across the dataset, anti-aliasing and normalization methods were first applied to the databases' image files. The Extended MNIST (E-MNIST) collection was created by reformatting these photos into a common 28x28 pixel size. This dataset has proven to be a great baseline for assessing the precision and accuracy of different machine learning character recognition models. This dataset is easily accessible by downloading it from the *Python Package Index* (PyPI), where it is available as separate packages such as letters, digits, by Class, by Merge, etc. For our use case, the letters section was separated, so it is different from the previously selected MNIST. As a result, character recognition models become more accurate and effective, and machine learning as a whole sees an increase in creativity, as suggested in **Fig. 7**.

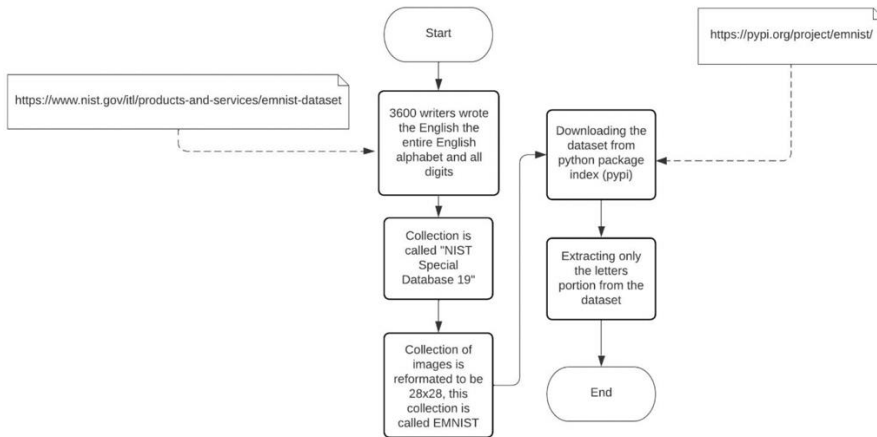


Fig. 7. Data Collection Visualization for E-MNIST dataset

3.2.3 Fashion-MNIST

Fashion-MNIST is a popular dataset that is a variant of the basic MNIST dataset. It was developed to serve as a more difficult and realistic benchmark for evaluating machine learning algorithms in the fields of computer vision and pattern recognition. Fashion-MNIST is a collection of grayscale photographs similar to MNIST, but, instead of handwritten digits, it includes images of clothing items. The dataset comprises 60,000 training photos and 10,000 test images arranged into 10 classes reflecting various fashion categories such as t-shirts, trousers, dresses, trainers, and more. When compared to MNIST, Fashion-MNIST produces a more realistic and diversified set of images, reflecting a greater range of objects and textures observed in real-world circumstances. Furthermore, Fashion-MNIST allows researchers as well as professionals to test the resilience and generalization capabilities of their algorithms beyond the basic job of digit recognition. The data collection for this dataset is presented in **Fig. 8**.

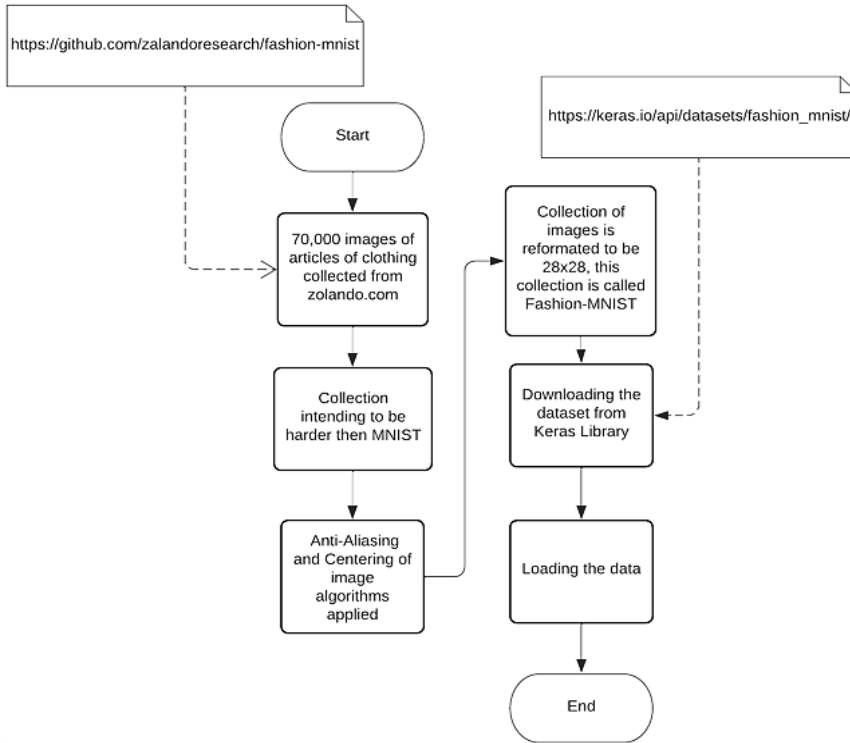


Fig. 8. Data Collection Visualization for Fashion-MNIST dataset.

3.2.3 MNIST-C

The MNIST-C dataset includes 15 corruption types applied to the MNIST test set. Its purpose is to serve as a benchmark for evaluating the out-of-distribution robustness of computer vision models. Through a series of experiments and visualizations, it has been observed that these corruptions significantly degrade the performance of state-of-the-art computer vision models while preserving the semantic content of the test images. The dataset consists of two components: the corruption matrix and the corruption severity levels. The corruption matrix defines the type and level of corruption applied to each image, while the corruption severity levels indicate the severity or intensity of the applied corruption. The MNIST-C dataset aims to capture a broad and diverse range of failure modes experienced by modern models.

Types of corruptions in an anomalous dataset are presented in **Fig. 9**.

- Shot noise: Randomly drop out some pixels from the image.
- Impulse noise: Randomly flip some pixels in the image.
- Gaussian noise: Add Gaussian noise to the image.
- Quantization noise: Quantize the pixel values to a small number of discrete levels.
- Contrast: Change the contrast of the image.

- Elastic distortion: Apply an elastic distortion to the image.
- Pixelation: Rescale the image to a smaller size and then upscale it back to the original size.

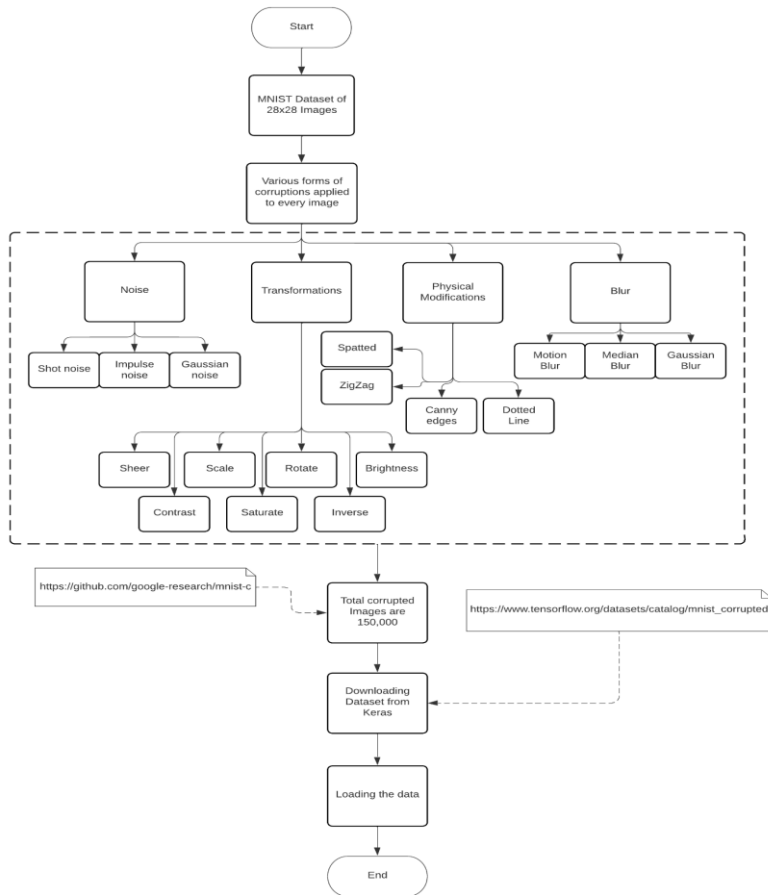


Fig. 9. Data Collection Visualization for MNIST-C dataset

3.2.4 K-MNIST

A set of images used to train machine learning models, particularly in the area of optical character recognition, is the *Kuzushiji-MNIST* dataset (OCR). The K-MNIST dataset contains 70,000 grayscale pictures, each of which represents a single Kuzushiji script character. As each picture is a square of 28x28 pixels, like the original MNIST dataset, it can be used with the current machine learning models and frameworks. Tasks involving anomaly detection may be successfully completed by using the K-MNIST dataset. It is appropriate for training machine learning models to recognize anomalies because of their originality, restricted exposure,

historical importance, data variety, and a sizeable dataset. The dataset may be preprocessed by using a conventional machine learning approach, and the majority of the K-MNIST samples can be used to train anomaly detection algorithms, like SVM or autoencoders. The abnormalities in fresh, undiscovered Kuzushiji characters can be found by using the taught models. Such metrics as accuracy, recall, and F1-score, can be used to assess the model’s performance. The accuracy of finding anomalous instances of Kuzushiji characters in a variety of applications can be improved by iteratively refining the anomaly detection models.

To construct the K-MNIST dataset, the following steps were undertaken as proposed in **Fig. 10**.

- **Data Collection:** A diverse array of historical documents containing Kuzushiji characters was gathered from various archives and libraries.
- **Annotation:** Experts proficient in the Kuzushiji script manually labeled and annotated the characters found within the collected historical documents. This involved identifying and assigning labels to individual Kuzushiji characters, corresponding to the modern Japanese Hiragana script.
- **Digitization:** The annotated historical documents underwent a digitization process, involving scanning or photographing, to transform them into digital image files.
- **Formatting:** The digitized and preprocessed images, along with their corresponding labels, were organized in a structured format compatible with prevalent machine learning frameworks and libraries.

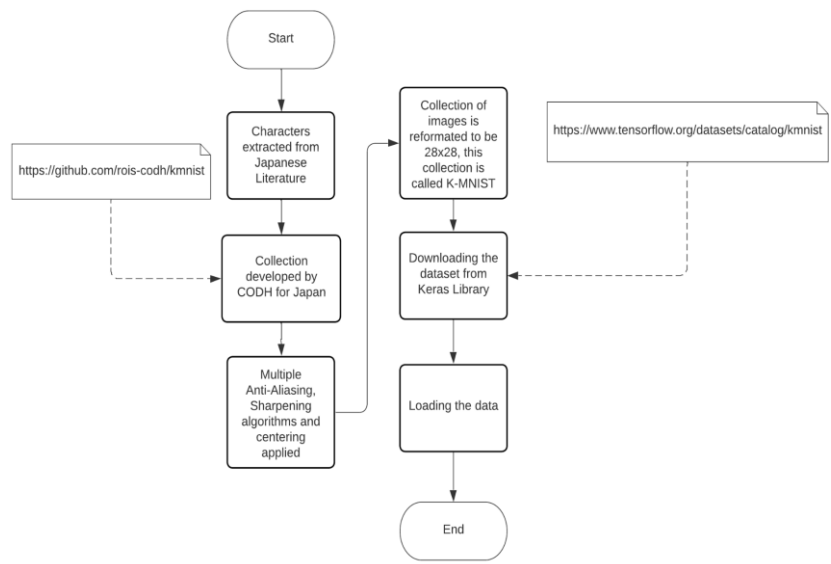


Fig. 10. Data Collection Visualization for K-MNIST dataset

3.3 Data Pre-Processing

Data pre-processing techniques are used to ensure the uniformity and suitability for training and testing machine learning models. The photos are resized to a consistent resolution, the pixel values are normalized to a standardized range, and the dataset is divided into training and validation sets. Preprocessing techniques serve to remove potential inconsistencies or biases in the data, thereby allowing for the best model performance. To perform pre-processing, it is frequently required to first load the dataset and then prepare it. The `tensorflow.keras.datasets.load_data()` function is a commonly used way to load datasets in Python. To guarantee that the data is in a format suitable for training machine learning models, many preprocessing methods are carried out after the dataset has been loaded, as suggested in **Fig. 11**.

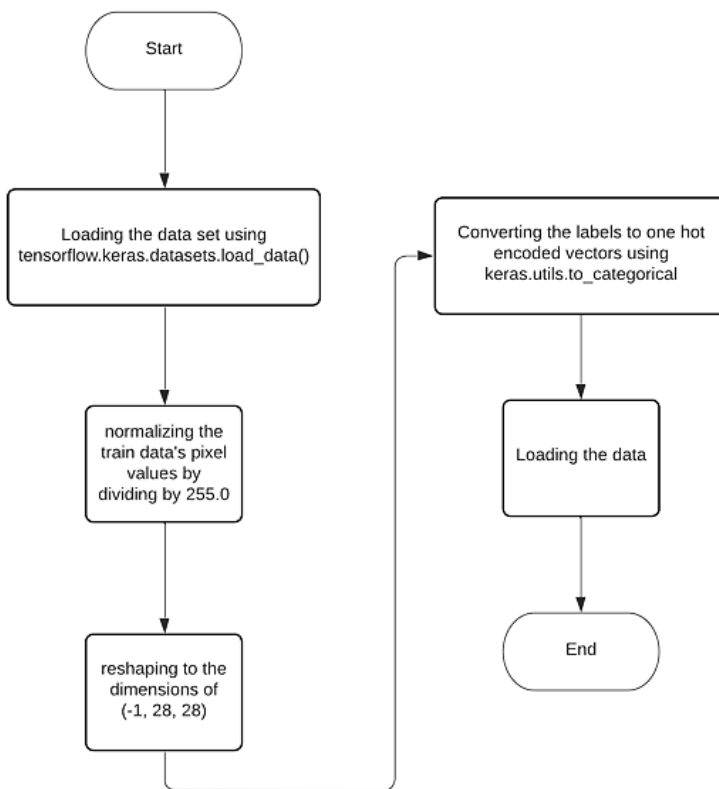


Fig. 11. Processing method for Data Pre-Processing

Since the pixel values in the MNIST, Fashion-MNIST, E-MNIST, MNIST-C and K-MNIST datasets range from 0 to 255, we scale the pixel values to a range of 0 to 1 by dividing them by 255.0. This can be accomplished easily by using Python's array division operator. The image data is reshaped to the appropriate dimensions after normalizing the pixel values. Although the photos in MNIST, Fashion-MNIST,

E-MNIST, MNIST-C and K-MNIST are all 28x28 pixels in size, data reshaping has dimensions of (-1, 28, 28), where -1 denotes that the size of the first dimension should be deduced from the sizes of the other dimensions.

Finally, the `keras.utils.to_categorical` method is used to transform the dataset's labels into one-hot encoded vectors. Working with classification problems frequently involves doing this since it enables the machine learning model to distinguish between the various classes with more ease.

3.4 Federated Learning

Federated learning is a machine learning technique which allows models to be trained by using decentralized data sources while maintaining data privacy. Data is often collected from multiple sources and centralized in a single area for training a model in conventional machines. Federated learning is a machine learning technique which allows for model decentralization while maintaining data privacy. Unlike the traditional machine learning systems which centralize data from several sources, federated learning involves an iterative model transfer procedure between a central server and distributed devices, resulting in enhanced model performance. **Fig. 12.** illustrates the proposed anomaly detection process of simple image classification. This presents the threshold control on pixel variance. The anomaly detection logic starts by asking whether the variance of the pixel values in the image exceeds a certain threshold. If 'yes', then the image is classified as an anomaly. If 'no', the anomaly detection logic moves on to check other features of the image. The anomaly detection logic then checks whether the image has sharp edges. If 'yes', the image is classified as an anomaly. If 'no', the anomaly detection logic checks the symmetry of the image. If the image is symmetrical, it is classified as a non-anomaly. If the image is not symmetrical, it is classified as an anomaly.

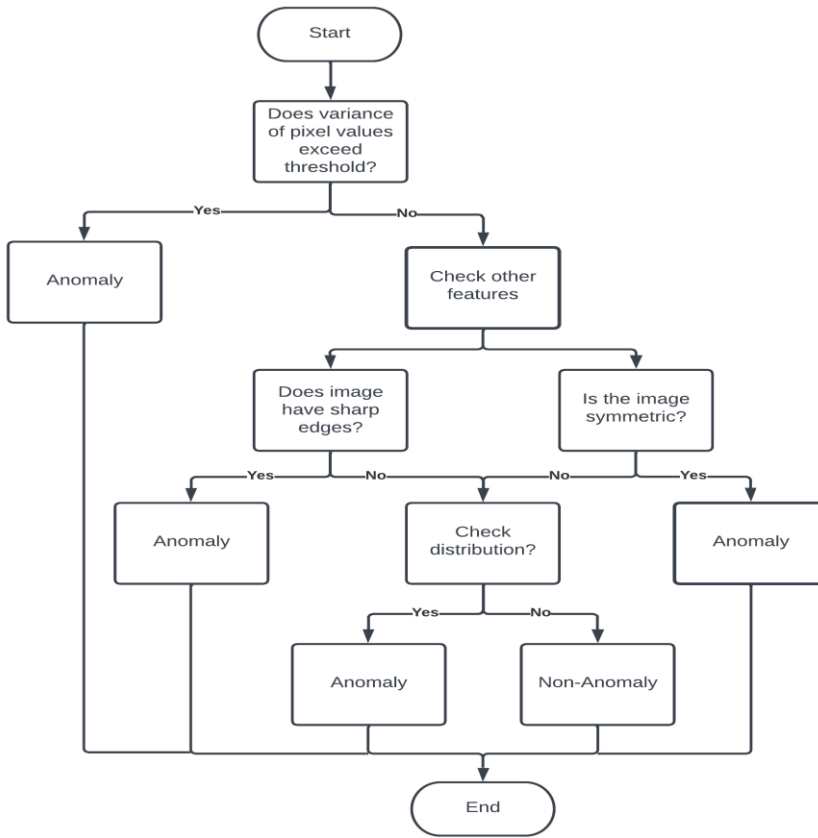


Fig. 12. Anomaly Detection Process Using FL Trained Model

The federated learning is used in this study to train a deep learning model while maintaining data privacy. The process entails sending an ‘initial model’ to connected customers who then train the model with their own data. Following that, the post-trained models or weights from each client are sent to a global server. To process the supplied models or weights, the server uses an aggregation function, such as FedAvg or FedAvgM. The complete procedure consists of numerous critical processes, including the weight transfer to the global server for updating and back-propagation execution. In this method, each round of communication is referred to as a communication round. This collaborative training strategy protects individual data while allowing for model improvement via iterative weight adjustments and increased communication. Retraining is done on a regular basis to keep the model up to date with the most recent data.

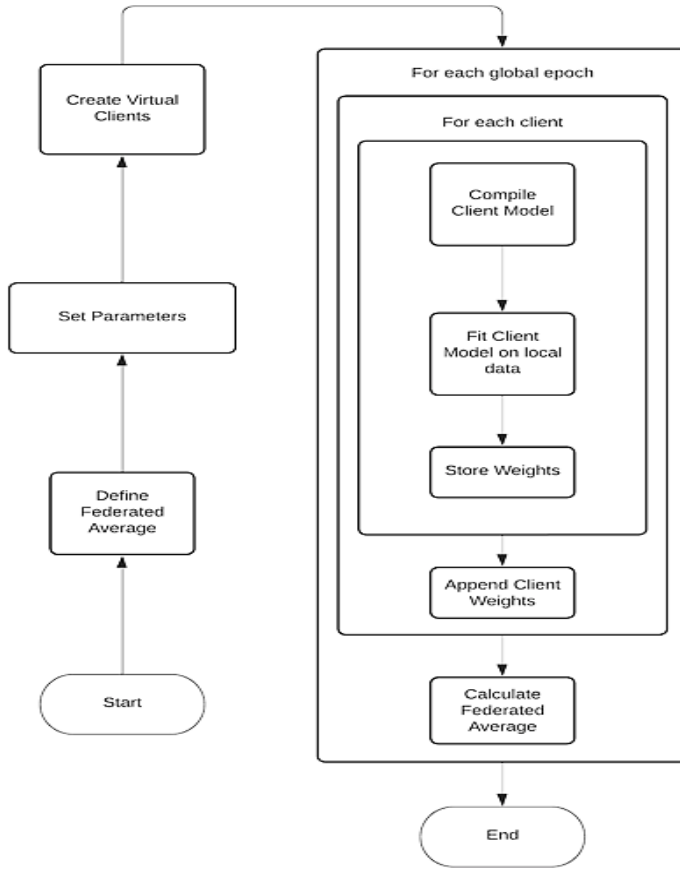


Fig. 13. Retraining Process of Federated Learning

Fig. 13. proposed the computation of weighted averages for model weights collected from numerous clients, while taking into account the number of clients, the client epochs, and the global epochs. A set of client models is constructed as clones of a pre-trained clustering model. The client models are iterated over for each global epoch, and each model is generated, fitted to the corresponding client's local data, and its weights are saved in a client weights list. To compute the average weights across all clients, the `federated_avg` function is used. These average weights are used to update the pre-trained clustered model. The revised model is tested, and the global epoch number and loss are displayed as performance metrics.

3.5 Anomaly Detection System

An anomaly detection system is a collection of approaches and algorithms that are designed to discover patterns or behaviors which deviate substantially from the expected or typical patterns within a given dataset. The importance of anomaly detection systems is derived from their ability to find early detection of deviation, fraud detection, fault detection, and to improve security. Anomaly detection algorithms used in our work are:

1. SVM (Support Vector Machine);
2. Auto-Encoder.

SVMs are supervised machine learning algorithms most renowned for their performance in classification tasks, but they can also be used for anomaly detection by utilizing their ability to define decision boundaries that separate different types of data. Autoencoders are a type of unsupervised deep learning models that can be used to detect anomalies. They are neural networks that are trained to learn a compressed version of the input data, or encoding, and then reconstruct the original input from this encoding. This method assists in capturing the underlying patterns and structure of the data, thereby revealing the data that deviates from the normal underlying patterns and exposing the anomalies.

3.5.1 Threshold Calculation

The word ‘threshold’ refers to a fixed amount or line used to assess whether a data point is considered normal or abnormal. There are 3 types of percentiles in threshold calculation. The first percentile denotes the lowest threshold values. Meanwhile, the third percentile denotes the highest threshold values. We used the 2nd percentile due to the moderate volume of our data, and thus we experimented with several numbers to see which could serve the best to distinguish between benign and anomalous data. Although the SVM model also recommended this number, it is a hyperparameter whose value may be adjusted to better suit our use case of anomaly detection. The quantile function value is specified as a constant (‘2’). In this manner, abnormalities in multiple datasets adjusted and configured the model.

Mathematically, anomalies score S for a test set. The threshold value t is such that all samples with an anomaly score below t are considered normal, and all samples with an anomaly score equal to or greater than t are considered anomalous. To find t , we first calculate the 2nd percentile of the anomaly scores by using the `np.quantile()` function with a parameter value of 0.02, as suggested in Equation (1).

$$\text{threshold} = \text{np.quantile}(S, 0.02) \quad (1)$$

$$\text{anomalies} = \text{np.where}(S \geq \text{threshold})[0] \quad (2)$$

$$\text{non_anomalies} = \text{np.where}(S > \text{threshold})[0] \quad (3)$$

Equations (2) and (3) present np. where () function returns an array of indices where the given condition is true. In this case, anomalies contain the indices of all samples in S that have an anomaly score less than the threshold, while non-anomalies contain the indices of all samples in S that have an anomaly score greater than or equal to the threshold.

3.5.2 Support Vector Machine

The Support Vector Machine (SVM) model is applied to pictures of handwritten digits from the MNIST dataset by using the Scikit-learn package. The train-test-split function of Scikit-learn is used for the purpose of splitting the data into the test and train sets. After the dataset has been loaded and divided into the training and test sets, a SVM model with a linear kernel and a regularization parameter C set to 1.0 is created by using the SVM.SVC class, and the model is trained on the training set using the fit () method. Following the training, the model is used to predict the test set's labels by using the predict () function, and its classification accuracy is calculated by using the score () method. This accuracy score reflects the model's ability to predict the labels of the test set. The trained SVM model is then used in the code to find anomalies in the test set. Finding data points that dramatically vary from the dataset's typical behavior is the process of anomaly identification. As shown in **Figs. 14,15,16,17** and **18**, anomalies are identified based on the decision function output of the SVM model which calculates the separation between each test data point and the model's decision borders for the datasets used in this work.

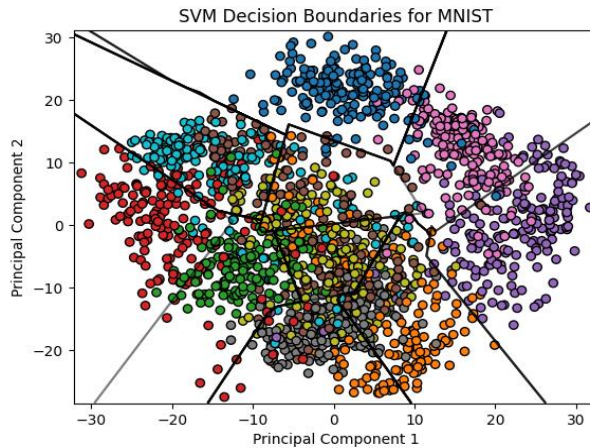


Fig. 14. SVM separates classes for MNIST dataset using clear boundaries

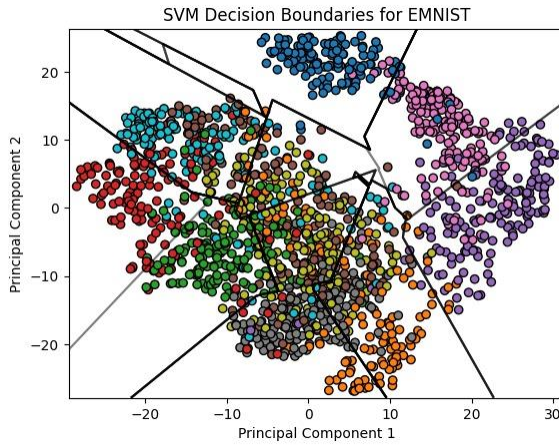


Fig. 15. SVM separates classes for E-MNIST dataset using clear boundaries

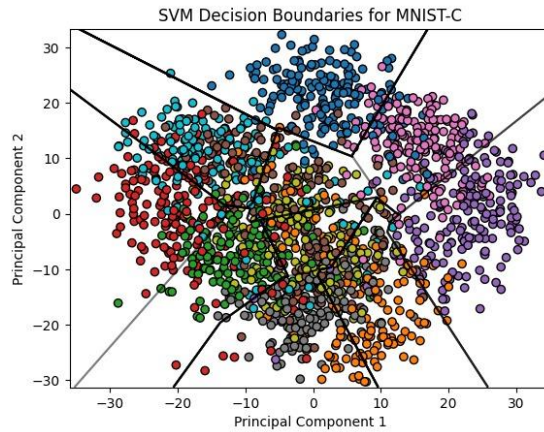


Fig. 16. SVM separates classes for MNIST-C dataset using clear boundaries

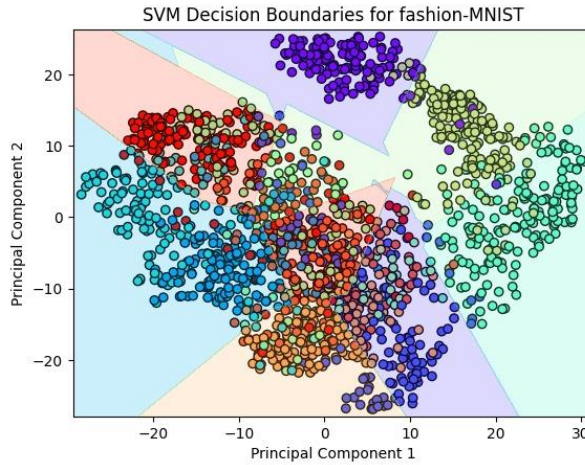


Fig. 17. SVM separates classes for Fashion-MNIST dataset using clear boundaries

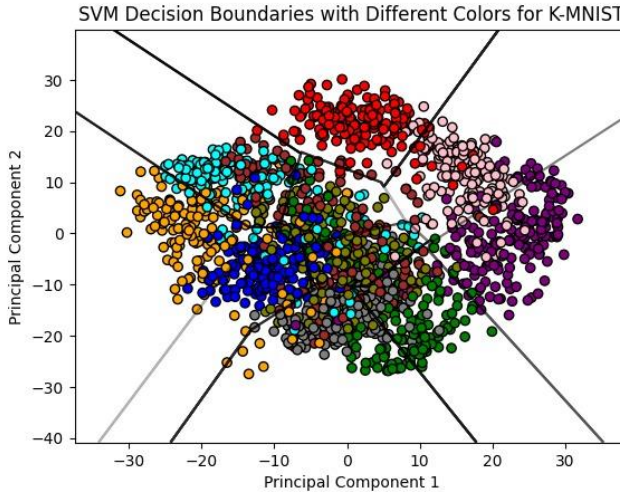


Fig. 18. SVM separates classes for K-MNIST dataset using clear boundaries

The `np.where()` function from the NumPy library is used to locate the indices of the test data points whose decision function output is below the threshold after setting a threshold value to zero in order to discover anomalies. The anomalies variable, which we can save as images or csv, has these indices that indicate the presence of anomalies. Anomalous tests and anomalous data are eventually merged with anomalous data as mixed data once the model has been trained and the reconstruction error set has been set. The trained model will be evaluated by using the mixed data values that were imputed. The loss values from the mixed data values will be returned, and they will be contrasted with the threshold values.

The input is recognized as anomalous data if its value exceeds the threshold values. If not, the input is recognized as innocuous data. All nine devices are used in this technique with the same local model structure as results from non-FL models. The datasets utilized in this study are non-identical, heterogeneous independent distribution datasets (non-IID). However, the FL models whole setup is dependent on how the client's data classes and features are distributed. The entire training process and accuracy may be impacted. We also suggest two methods for client selection and retraining so that to enhance the functionality of the FL model. The Results section demonstrates that the client selection and retraining technique, whether used or not, has various effects on the effectiveness and robustness of the FL model.

3.5.3 Auto Encoder

A typical form of an artificial neural network used for dimensionality reduction and unsupervised learning applications is the autoencoder. Without the need for specific labeling, it is intended to learn effective representations or compressed copies of the input data. An encoder and a decoder are the two primary parts of an autoencoder's construction. The encoder extracts the most significant characteristics or patterns from the input data and maps them onto a lower-dimensional latent space. The original input data is subsequently recreated by the decoder using the encoded representation.

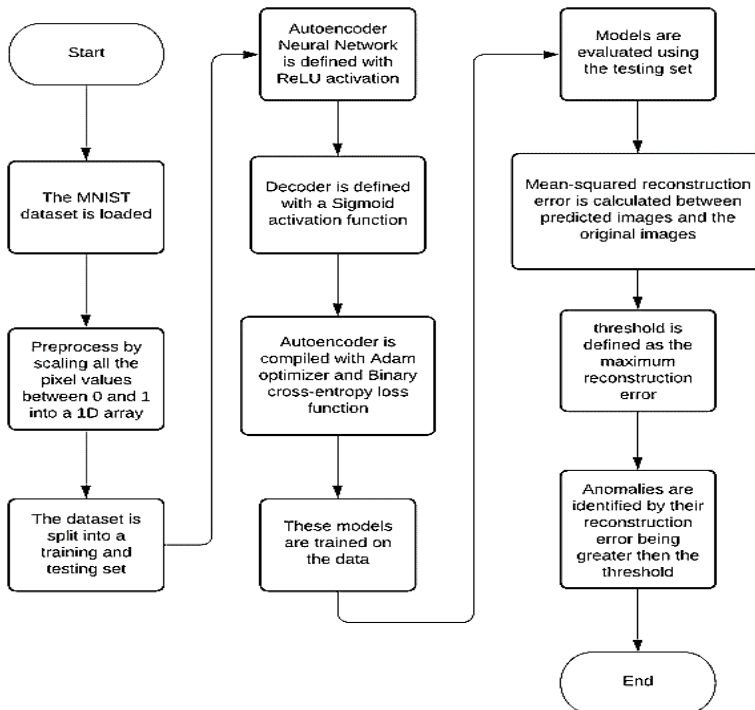


Fig. 19. Methods for Anomaly detection for Auto Encoder

To further the attempt of detecting the anomalies, an autoencoder was used in this work. **Fig. 19.** suggested the MNIST dataset prepared for the use with an autoencoder neural network by flattening the pictures into a 1D array and scaling the pixel values in each image between 0 and 1. A training set and a testing set were then created from the dataset. The training set is used to train the autoencoder model, and the testing set is used to assess its effectiveness. The autoencoder neural network is created by using the ReLU activation function, while the decoder neural network is created by using the Sigmoid activation function. Afterwards, the autoencoder is built, followed by the Adam optimizer and binary cross-entropy loss function, which are often used in neural network training. By adjusting the learning rate for each parameter during training, it combines the benefits of two other optimizers, AdaGrad and RMSProp. In comparison to the conventional gradient descent methods, this leads to faster convergence and improved generality. The mean-squared reconstruction error, a measurement of how effectively the autoencoder can reconstruct the input data, is calculated between the predicted images and the original images. When categorizing data into one of two groups, binary classification tasks frequently employ the loss function known as binary cross-entropy (BCE) loss (e.g. normal or anomalous). The BCE loss calculates the discrepancy between the genuine class labels and the projected probabilities, which results in a larger penalty for incorrect classifications. By contrasting the reconstructed output with the original input, the BCE loss is used to train autoencoders in the context of anomaly detection. The autoencoder's objective is to reduce the reconstruction error, which is quantified by the BCE loss. When employing auto-encoders for anomaly detection, anomalies are found by first training the encoder to learn a representation of the input data and then analyzing the reconstruction error. When employing auto-encoders for anomaly detection, anomalies are found by first training the encoder to learn a representation of the input data and then analyzing the reconstruction error. A description of the procedure along with the relevant math equations is presented below:

Auto-encoder trainer

$$\text{Input data} = X_{Rn} \quad (4)$$

where: n is number of features

$$\text{Encoder function } f(X) = W_1 X + b_1 \quad (5)$$

$$\text{Latent representation (Z Rm): } g(Z) = W_2 Z + b_2 \quad X' = g(f(X)) \quad (6)$$

where: m is the number of latent variables.

$$\text{Reconstruction error Error.. } E(X, X') = ||X - X'||^2 \quad (7)$$

Equation 7 is used to calculate the reconstruction error, which is the difference between the original input X and the reconstructed input X' . The error is calculated by

using the Euclidean distance between the two inputs. The user suggests setting a threshold for detecting abnormalities by recognizing anomalies when the expected value of X is greater than the threshold T . We thus set a threshold for identifying abnormalities as Equation 8 suggests below.

$$\begin{aligned} &\text{Threshold for anomaly detection: } T \\ &\text{such that } E(X, X') > T \quad \square \quad X \text{ is an anomaly.} \end{aligned} \quad (8)$$

3.6 Zero Knowledge Proof

In the realm of anomaly detection, ZKP can be used to verify the accuracy and legitimacy of detection algorithms without exposing the underlying data or compromising user privacy. By leveraging ZKP, organizations can create secure and privacy-preserving anomaly detection systems that can be audited and verified by external parties. ZKP to detect anomalies using a statistical approach, such as the Z-score, while maintaining anonymity.

$$Z = (X - \mu) / \sigma \quad (9)$$

The Z-score is represented by Z , while the data item is represented by X . Equation 9 presenting the mean of the dataset is the standard deviation of the dataset. To use ZKP for privacy-preserving anomaly detection, Z-score calculation was performed correctly without revealing any information about x , μ and σ .

The prover creates a set of data X and uses an algorithm for anomaly identification to produce a set of anomalies S . Statement P defined by Prove denotes the quality of S being a legitimate collection of anomalies. P , for instance, read, “ S has at most k anomalies,” where k is a preset cutoff. A random number r is generated by the prover, and then the commitment is calculated.

$C = H(r, P)$, using the cryptographic hash function H .

The prover sends C to the verifier, along with the set S . The verifier chooses a random challenge value c and sends it to the prover as suggested in Equation (10) below.

Response value computation by the Prover

$$s = r + c * f(S) \quad (10)$$

where: $f(S)$ is a function that maps S to a unique value; r is a random number in F ; c is a random challenge value in F ; and $C = H(r, P)$ is the commitment; “ $s = r + c * f(S)$ ” is the response value.

The prover sends s to the verifier. The verifier checks the value of C and s . The verifier determines that S is a valid collection of anomalies if both checks are successful without knowing anything other than the knowledge that S meets the condition P . If both checks pass, the verifier S is a valid set of anomalies without

learning anything beyond the fact that S satisfies the property P . This not only enhances the trustworthiness of the detection models but also ensures that they are compliant with stringent data privacy regulations, such as the *General Data Protection Regulation* (GDPR).

3.7 Blockchain

Blockchain technology has the potential to revolutionize the way data is stored and managed, by providing a secure, decentralized, and transparent platform for data sharing and validation. By integrating Blockchain into anomaly detection systems, organizations can ensure that their data is tamper-proof and that the detection algorithms are trusted and verified. Blockchain's immutable ledger guarantees the integrity of the data and the detection models, thus making it virtually impossible for malicious actors to manipulate the system. Moreover, the decentralized nature of Blockchain technology eliminates the need for centralized servers, thus reducing the risk of single points of failure and improving the overall security of the system. Regarding the anomaly detection process using blockchain, let $X = X_1, X_2, \dots, X_n$ be a dataset to be monitored for anomalies. Define a hash function H that can take any data point X_{ij} and produce a unique hash value $h(X_{ij})$, a block B is created that contains the hash values of all data points in X , such that $B = h(X_1), h(X_2), \dots, h(X_n)$. Each time a new data point X_{new} is added to X , compute its hash value $h(X_{new})$ and add it to block B . Define an anomaly detection function $F(B)$, which takes block B as the input and returns a binary value indicating the presence or absence of an anomaly.

Deploy the smart contract SC that monitors block B for any anomalies: $SC(B) = F(B)$ each time the block B is updated, A transaction Tx is created that includes the updated block B and the output of the anomaly detection function $F(B)$: $Tx = (B, F(B))$. The transaction Tx is broadcast to the network and is validated by the network nodes. This is the method for creating a block B that contains the hash values of all data points in X , with a function $F(B)$ for detecting anomalies by using blockchain.

3.8 Supervised Learning Framework

Supervised learning is a popular approach in machine learning that involves training a model on labeled data. This means that the data used to train the model is already labeled with the correct output, thus allowing the model to learn to make accurate predictions based on the input features. The process of supervised learning typically involves splitting the labeled data into the training and testing sets. The training set is used to train the model, while the testing set is used to evaluate its performance. Federated learning reduces the amount of data that needs to be transmitted between devices. Since the model is trained locally on each device, only the updates to the model need to be transmitted back to the central server. Blockchain is used in supervised learning in several ways. By storing the data on a

blockchain, it is possible to ensure that the data remains private and secure. Each block in the chain contains a hash of the previous block, and it is easy to verify the integrity of the data. The benefit of using blockchain in supervised learning is that it can increase transparency. Since the ledger is distributed across multiple nodes, it is easy to see who has accessed the data and when they did so. Finally, blockchain can also reduce the computational burden on the central server. Let us assume a dataset with n samples and k features. A sample is represented as a vector explained in Equation (11).

$$x_i = [x_{i1}, x_{i2}, \dots, x_{ik}], \quad (11)$$

here: x_{ij} is the j^{th} feature of the i^{th} sample. We train a binary classifier by using this dataset, where the output variable y_i is 1 if the sample x_i is anomalous, and 0 otherwise. The input to the classifier is the feature vector x_i . The usage of various classification algorithms, such as logistic regression, decision trees, random forests, support vector machines, or neural networks serves to build a model that can predict anomalies as Equation (12).

$$y = f(x) \quad (12)$$

here: $y = 1$ for anomalous x and $y = 0$ for normal x . x is the input feature vector, and f is the function that maps the input to the output. The function f can be learned by using a supervised learning algorithm which minimizes the prediction error on the training data. The model is trained to predict anomalies in new data by passing the feature vector of the new data to the model and obtaining the output label. If the output label is 1, then the new data is anomalous, and if it is 0, then the new data is normal. Anomaly detection is achieved by using federated learning and Blockchain in Equation (13) and the Federated Learning formula presented in Equation (14).

$$\text{Score} = p(x) / q(x) \quad (13)$$

here: $p(x)$ is probability density function of the data point. $q(x)$ is estimated probability density function of the data point, and in

Federated Learning formula:

$$\theta^* = \text{argmin} (\sum_{i=1}^N w_i F(\theta)) \quad (14)$$

here: θ^* = optimal parameters; w_i = weight assigned to the i th client; $F(\theta)$ = objective function; and N = the number of clients for Blockchain Formula.

$$G(n+1) = G(n) + \text{SHA-256}(\text{Block}(n) + G(n)) \quad (15)$$

here: $G(n+1)$ is the hash of the next block; $G(n)$ is hash of the previous block; $\text{Block}(n)$ is data in the current block; and SHA-256 is the cryptographic hash function which calculates the hash for every next block in the blockchain.

Equation (15) presents formulas combined to create a system that uses federated learning to train a supervised learning model to detect anomalies in data. The blockchain is used to securely store and share the model's parameters across multiple nodes in a decentralized manner. Zero-knowledge proof is used in supervised learning in several ways. By using zero-knowledge proof, it is possible to prove to the central server that a particular piece of information is correct without revealing the actual information itself. Let X be the input data, Y be the output data, and W be the weights of the model. The goal of supervised learning is to learn the function $f(X, W)$ which maps the input X to the output Y . With federated learning, multiple parties collaborate to learn the function $f(X, W)$ without sharing their private data. The parties train their local models on their own data and share only the model updates instead of the raw data. Therefore, the supervised learning approach using federated learning, blockchain, and zero-knowledge proof can be represented as Equation (16) which minimizes the sum of squared errors between the predicted output $f(X, W)$ and the actual output Y .

$$f(X, W) = \text{argmin}(W) \sum (y - f(X, W))^2 \quad (16)$$

here: $\text{argmin}(W)$ is the weight of the model. Zero-knowledge proof in supervised learning reduces the computational burden on the central server. Since the information is not revealed, it is possible to distribute the computation across multiple nodes, thus making it easier to scale up machine learning projects. **Fig. 20.** depicts the supervised learning approach by using Federated learning, Blockchain and ZKP Supervised learning. Here, a model is trained on labeled data to make predictions on new, unseen data.

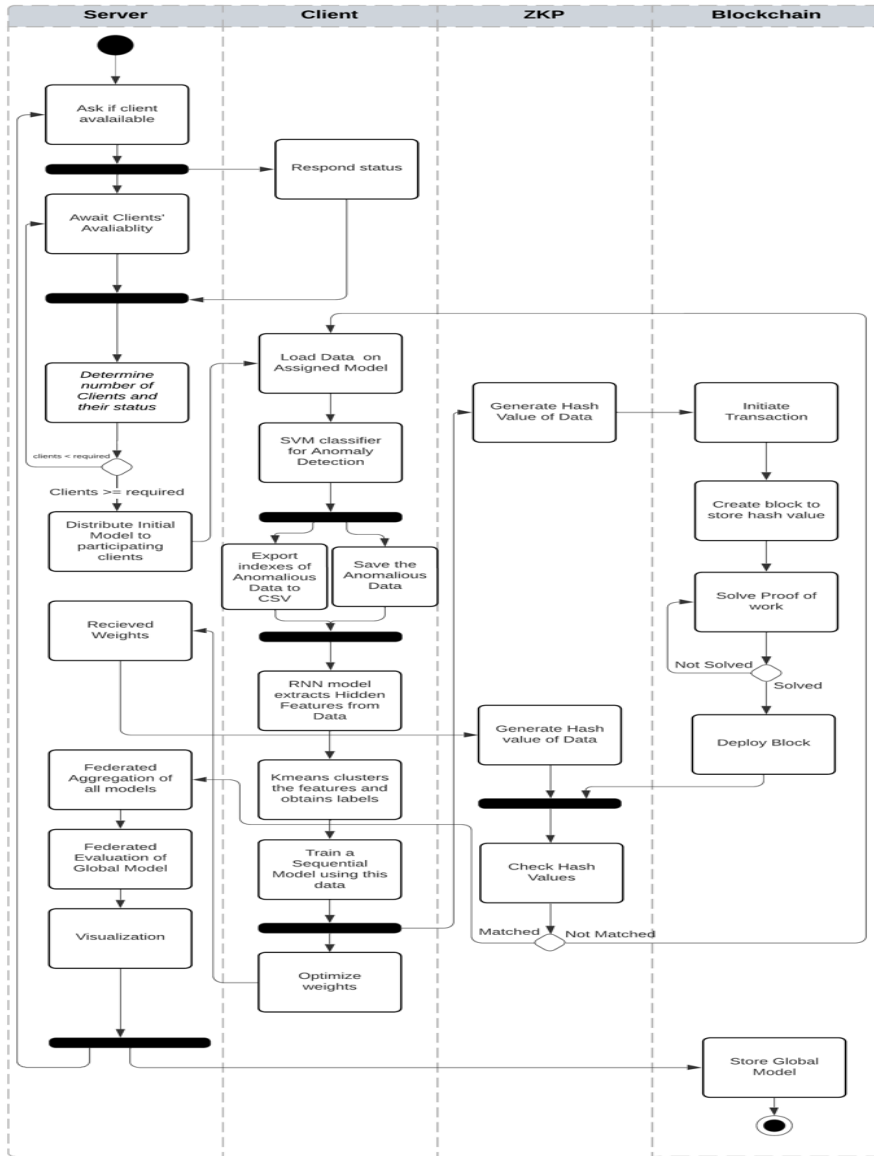


Fig. 20. Supervised Learning approach using Federated learning, Blockchain and ZKP

3.9 Unsupervised Learning Framework

Unsupervised learning is a machine learning approach in which the model is trained on unlabeled data, which means there are no predefined categories or classes. Unsupervised learning finds meaningful patterns and relationships in the data

without any prior knowledge of what the data represents. There are two main types of unsupervised learning algorithms: clustering and association. Clustering algorithms group similar data points together, while association algorithms identify patterns and relationships between variables as presented in Equations (17), (18).

$$\text{Anomaly Score} = (1 - \alpha) \times \text{Local Anomaly Score} + \alpha \times \text{Global Anomaly Score} \quad (17)$$

here: α is a weighting factor between 0 and 1; α determines the balance between local and global anomaly scores.

$$\text{Local Anomaly Score} = |X - \mu| / \sigma \quad (18)$$

here: X is the input data point; μ is the mean of the local data distribution; and σ is the standard deviation of the local data distribution.

$$\text{Global Anomaly Score} = |X - \mu'| / \sigma' \quad (19)$$

here: μ' is the mean of the global data distribution; σ' is the standard deviation of the global data distribution.

Equation (19) calculates the global anomaly score. This score measures how far away the input data point is. Blockchain is a decentralized ledger that enables secure and transparent data sharing, and zero-knowledge proof is a cryptographic technique that enables a party to prove possession of certain information without revealing that information. The formula for unsupervised learning using federated learning, blockchain, and zero-knowledge proof can be expressed as minimized loss suggested in Equation (20).

$$\theta \times (\sum_{n=1}^i f_i(\theta)) \quad (20)$$

here: θ is the model parameter; n is the number of participant devices or servers in the federated learning; and $f_i(\theta)$ is the local loss function for the i th device or server.

The objective of the formula is to minimize the total loss function across all devices or servers without compromising the privacy of the data. The combination of federated learning, blockchain, and ZKPs can enable secure and privacy-preserving unsupervised learning as proposed in **Fig. 21**.

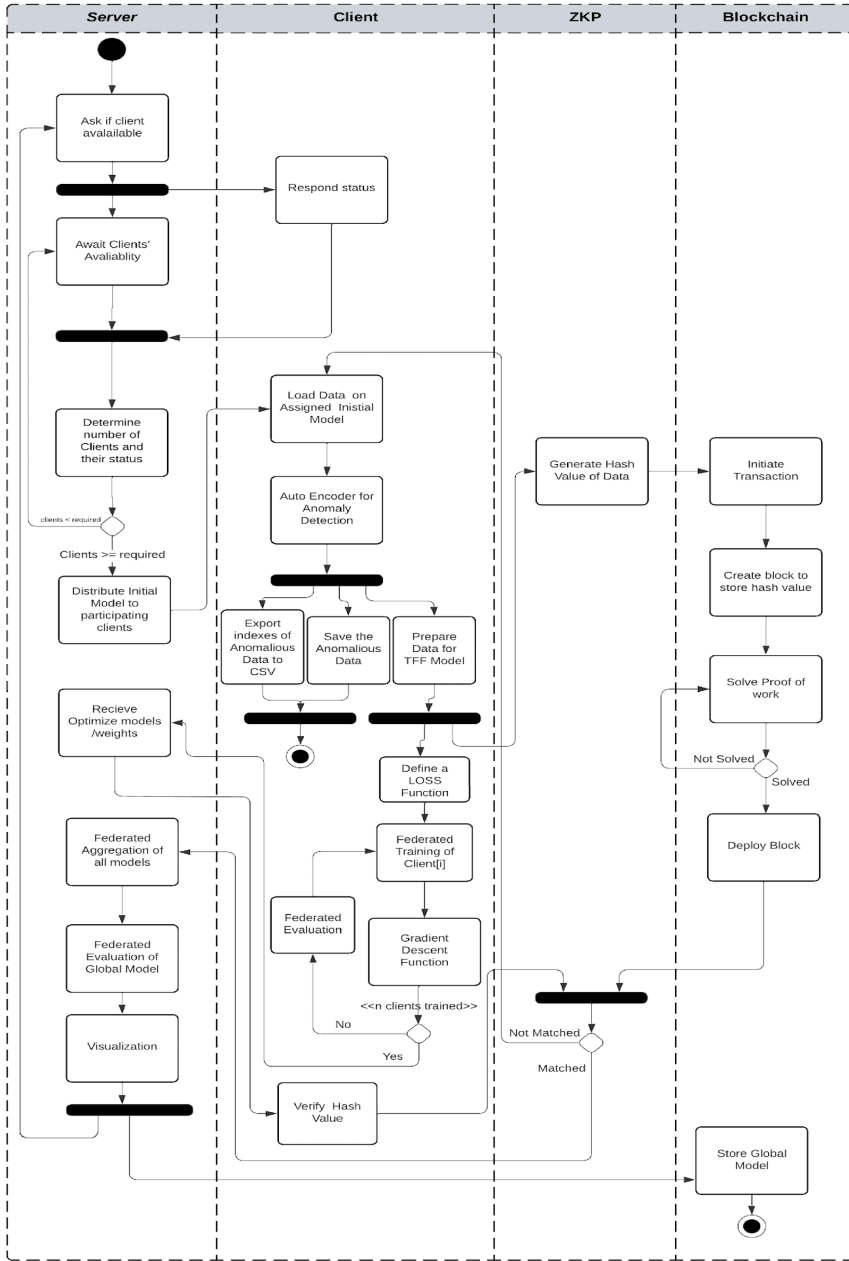


Fig. 21. Unsupervised Learning approach using Federated learning, Blockchain and ZKP

3.10 Application Framework

A framework for detecting anomalies using zero-knowledge proofs (ZKP), federated learning, and blockchain combines the strengths of these three technologies to create a secure, private, and decentralized system for identifying irregularities in datasets. The forward process, transmitting weight to the global server for the update, and the back-propagation process are the three main steps in the entire implementation. The entire process is referred to as one communication loop in federated learning. Because the weights have been modified over multiple communication rounds, the federated model performs better when there are more communications. The federated model design also includes a retraining process which allows the model to work with non-IID datasets, thus increasing the model's usefulness in real-world applications. Prior to the weights being aggregated on the global server, the clients' models would go through additional training as part of the retraining process. Each entity involved in the anomaly detection process would contribute data to a federated database. After that, ZKP methods would be used to train the model to make sure that no party's data has been revealed. ZKP is used to validate computations during the anomaly detection process without revealing any sensitive data. Once trained, the model is used to spot irregularities in real-time data streams. Both supervised and unsupervised methods are applied in the framework. The supervised technique enables several parties to work together without sharing data since labeled data can be used to train a model using federated learning. An immutable and transparent blockchain network can be used to store the trained model. On the other hand, unsupervised anomaly detection can be performed by using ZKP, which allows the parties to prove the validity of their data without revealing it. This approach can be beneficial in scenarios where data privacy is crucial. Each entity involved in the anomaly detection process would contribute data to a federated database. **Fig. 22.** proposes a framework of basic components suggested for anomaly detection.

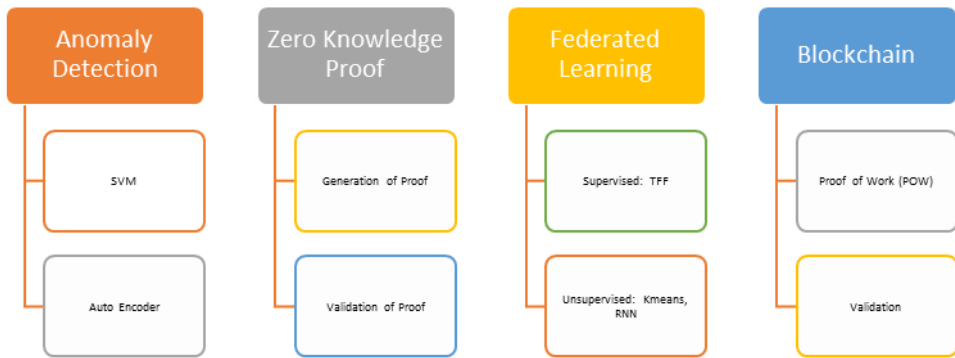


Fig. 22. Proposed framework basic components

Both supervised and unsupervised methods are applied in the framework. Federated learning is a machine learning technique which allows multiple clients to work together on a shared model while keeping their data private. **Fig. 23.** proposes the process of federated learning, starting with the server determining the number of connected clients and randomly assigning them as supervised or unsupervised.

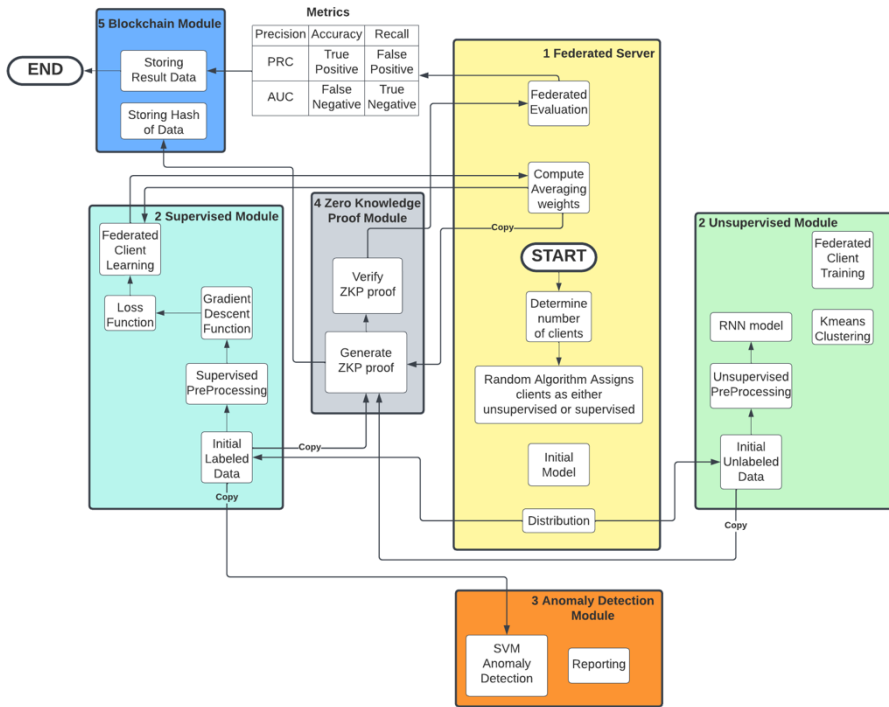


Fig. 23. Framework for Anomaly detection using ZKP, FL and blockchain using Supervised and Unsupervised approaches

The unsupervised clients preprocess their unlabeled data, extract hidden features by using an RNN model, cluster the data with the KMeans algorithm, and train their copy of the initial model. A hash of the weights derived from the unlabeled data is generated by using the ZKP module and stored in the blockchain. The supervised clients preprocess their labeled data, define a gradient descent function with a loss function, and train their copy of the initial model. A hash of the weights derived from the labeled data is generated by using the ZKP module and stored in the blockchain. The server computes the hash of the received weights, verifies it with the previously stored hash, and computes the averaging weights. These average weights are then sent back to modify the initial local model which is then retrained on the new weights generated by the clients.

Federated evaluation takes place, and the metrics data is stored on the blockchain. This cycle continues for a set number of epochs, thereby allowing the model to improve over time without compromising the privacy of the client's data. **Table 9.** illustrates the whole concept of a framework for anomaly detection using ZKP, FL, and blockchain using the supervised and unsupervised approaches.

Table 9. Module Explanation for Anomaly detection using ZKP, FL and blockchain using Supervised and Unsupervised approaches

Module	Explanation
Server	• Determines the number of clients that are connected to the server. • Random Algorithm Assigns them as either unsupervised or supervised depending on factors such as the data they have and their computational capabilities. • The initial model is distributed to all the clients.
Anomaly Detector	• The data in each of the clients is passed through an anomaly detector which detects anomalies according to the then-calculated threshold value. This threshold separates the normal from the anomalous data. • The detected anomalies can be stored as a csv file.
Unsupervised Clients	• The unlabeled data is preprocessed to fit into the model. • The RNN model extracts the hidden features. • The KMeans algorithm clusters the data with the hidden features. • The Client trains its copy of the initial model on this clustered data. • A hash of the weights derived from the unlabeled data is generated by using the ZKP module. The hash is stored in the blockchain for further use. • The Client sends the server its trained list of weights.
Supervised Clients	• The labeled data is preprocessed to fit into the model. • We declare and define a gradient descent function with a loss function to do the calculations. • The Client trains its copy of the initial model on this data by using the functions. • A hash of the weights derived from the labeled data is generated by using the ZKP module.

	• The hash is stored in the blockchain for further use. • The Client sends the server its trained list of weights.
ZKP verification module	• If it matches, then the execution continues. • Else: the weights are discarded, and execution moves to the next epoch. • The Server computes their averaging (aggregation) weights and then sends back these average weights to modify the initial local model. • The model is retrained on the new weights that are generated from the clients which are using the updated initial model. • Federated Evaluation takes place. • Metrics are generated from this evaluation data. • The metrics data is then stored on the blockchain. • This cycle continues according to the set number of epochs.
Blockchain	• The model is stored on blockchain. • We generate the hash value of data. • We initiate a Transaction. • We create a Block. • We store the hash value. • We solve Proof-of-Work. • We deploy the block.

3.11 Section Summary

Anomaly detection is a crucial task that is becoming increasingly challenging due to the dynamic nature of modern data, technological advancements, and vast and heterogeneous data. Traditional anomaly detection systems fail to address the requirements of modern data, such as scalability and adaptability, and struggle to process and analyze big and diverse datasets efficiently. To address these challenges, this chapter proposes a unique and multifaceted approach that combines federated learning, zero-knowledge proofs, and blockchain technology. The integrated

technology offers immense real-world value and can transform anomaly detection methods. The proposed approach ensures privacy, integrity, and accountability while increasing scalability, adaptability, and interpretability.

4.EXPERIMENTAL STUDIES

4.1 Overview

In this section, we define a federated learning model and a smart contract for the blockchain network using the Syft and SmartPy libraries. The model is trained by using decentralized data, and the model weights that result from it are encrypted by using zero knowledge proof techniques. These encrypted weights are then securely saved on the smart contract in order to forecast anomalies in an efficient and trustworthy manner. The encrypted model weights are used to predict anomalies as part of the process. The predictions are decrypted by using zero knowledge proof, and the results are shown on the console. The goal of this experimental project is to provide useful insights into the performance and feasibility of using federated learning for anomaly detection tasks. The findings of this study will help to further extend the existing research efforts in privacy-preserving machine learning and provide practical guidelines for constructing anomaly detection systems in a distributed and secure manner. This effort intends to advance the field of anomaly detection while preserving data privacy and security by harnessing the power of federated learning and zero knowledge proof.

4.2 Experimental Setup

Resources

The following settings as presented in **Table 10**. were used for all tests in the Jupyter Notebook virtual cloud environment.

Table 10. Experimental environment setup

Hardware	Configuration
CPU	Intel Celeron CPU @ 1.80Ghz
Hard Disk	500GB
Memory	8GB
GPU	Intel ® HD Graphics

The *PySyft* libraries were constantly used in this project for safe, private deep learning. To preprocess data and provide the assessment metrics, we mostly use scikit-learn tools. The local models and Federated Learning models were constructed by using *Pytorch*.

4.3 Supervised Learning Approach to Detect Anomalies by Using Zero-Knowledge Proof along with Federated Learning and Blockchain.

Anomaly detection is an important problem in many fields, including finance, cybersecurity, and defect identification. The traditional anomaly detection approaches frequently necessitate centralized access to sensitive data, thus raising privacy problems. Federated learning, which allows collaborative model training without revealing raw data, is a possible approach. The purpose of this experimental project is to investigate the use of federated learning for anomaly detection. The primary goal of this research is to explore the efficacy of a federated learning strategy in finding anomalies in a given dataset. The research covers training an SVM model for anomaly identification, creating visual representations of anomalies, preparing data for a *TensorFlow Federated* (TFF) model, and evaluating the performance of the federated learning approach.

4.3.1 Experimental Steps

Loading the dataset into memory: The first step is to load the dataset into the memory. This phase entails reading the dataset from a file and storing it in an appropriate data structure for subsequent processing. The chosen data format would be determined by the nature of the dataset and the machine learning technique being used. A Pandas data frame, for example, can be used to hold a tabular dataset.

The next stage is to detect anomalies with the *Support Vector Machine* (SVM) technique. This entails utilizing the dataset to train an SVM model and then use it to detect abnormalities. The SVM model's hyperparameters and kernel functions can have a substantial impact on the performance of the anomaly detection technique.

After discovering anomalies, the task is to create a CSV file containing the abnormal data. This can be accomplished by extracting the anomalous data from the original dataset and saving it as a new CSV file. After that, the CSV file can be utilized for further analysis or visualization.

The next stage is to extract the anomalous data from the CSV file and create pictures to illustrate the anomalies. This can be accomplished with a visualization library, such as *Matplotlib* or *Seaborn*. The graphics can aid in the identification of patterns and trends in anomalous data.

The next stage is to create zero-knowledge proofs (ZKPs) from the preprocessed data. This entails creating cryptographic proofs that can be used to validate the data without revealing the real data. ZKPs can aid in the protection of data privacy and security during the federated learning process.

Then, it is necessary to define a suitable loss function for the federated learning model. The loss function which is utilized will be determined by the nature of the problem and the machine learning technique in use. The mean squared error, cross-entropy loss, and hinge loss are examples of common loss functions.

The next step is to train the federated learning model on each client device. This entails transmitting the preprocessed data and ZKPs to each client device and training the model locally with the data from the client. The modified model parameters are then sent to the server for aggregation by the client devices.

A gradient descent function is utilized throughout the training process to update the model parameters depending on the local data on each client device. The gradient descent function can be implemented with an optimization library like *TensorFlow* or *PyTorch*. After each client has completed training, the model's performance on each client device is evaluated. This entails such computing measures as accuracy, precision, and recall on local data from each client device.

Before delivering the revised model parameters to the server, the ZKPs generated in Step 6 are checked to confirm that the data was not tampered with or modified throughout the federated learning process.

Step 1: The learning rate is set to 0.1, and the zero-knowledge proof validation is initially set to 'initial_model'.

Step 2: The number of rounds is set to 5, and the loss and learning rates are saved in separate lists. The data is transformed to a Panda Data Frame 'df' before the training process begins.

Step 3: The model is trained with the learning rate and federated train data using 'federated_train'. If the number of rounds is two, the noise value is one; otherwise, it is zero.

Step 4: The proof is validated by using 'verify_proof', which evaluates the proof's validity given the column values and random values.

Step 5: If the proof is correct, the learning rate is multiplied by 0.9, and the loss value is computed with 'federated_eval'. The round number and the loss value are then printed after the loss value and learning rate are attached to their respective lists.

Step 6: If the proof is invalid, the batch is skipped, as stated in Algorithm 1, and **Table 11.** displays all of the results we have obtained from implementing ZKP validation. The Client server response on each round gives the loss numbers.

Algorithm 1. ZKP validation

```

1. model = initial_model
2. learning_rate = 0.1
3. number_of_rounds = 5
4. loss_values = []
5. learning_rates = []
6. df = pd.DataFrame(federated_train_data)
7. for round_num in range(0,number_of_rounds):
8.   model = federated_train(model, learning_rate, federated_train_data)
9.   if
10.    number_of_rounds == 2:
11.    noise = 1
12.  else:
13.    noise = 0
14.  if
15.    verify_proof(df, column_values, proof, random_values, noise):
16.    print("ZKP VALIDITY PASSED, CLIENT -> Seerver IS SECURE")
17.    learning_rate = learning_rate * 0.9
18.    loss = federated_eval(model, federated_train_data)
19.    loss_values.append(loss)
20.    learning_rates.append(learning_rate)
21.    print('round {}, loss={}'.format(round_num, loss))
22.  else:
23.    print("ZKP VALIDITY CHECK FAILED, skipping this batch.

```

Table 11. ZKP Validation

ZKP VALIDITY	CLIENT	ROUND	LOSS
PASSED	Server IS SECURE	0	20.691389083862305
PASSED	Server IS SECURE	1	19.16118049621582
PASSED	Server IS SECURE	2	17.984769821166992
PASSED	Server IS SECURE	3	17.064706802368164
PASSED	Server IS SECURE	4	16.326141357421875

If the ZKPs are valid, the client devices can securely transmit their updated model parameters to the server. To preserve the privacy and integrity of the data, this communication procedure should be encrypted. Federated evaluation of weights is calculated by the server. After receiving the updated model parameters from the client devices, the server executes federated evaluation by aggregating the weights from each client. Depending on the federated learning framework being utilized, this aggregation can be accomplished by using approaches such as federation averaging or weighted averaging.

Otherwise, we skip the batch of training data. If the ZKPs are discovered to be faulty during the verification phase, this implies manipulation or malicious behavior. In such circumstances, the server skips the related client's batch of training data and moves on to the next batch.

Finally, we record the outcomes on the blockchain. The findings of the federated evaluation can be kept on a blockchain for transparency, immutability, and auditability. Storage of the results on a blockchain ensures that they cannot be changed or manipulated after the fact, thereby providing a verifiable record of the federated learning process and its outcomes, as presented in **Table 12**.

Table 12. Blockchain verifiable records

Index	Timestamp	Data	Proof	Hash value
2	1679842660. 113144	5	3529 3	[d2b3d509682326c7d968f49beed8 5ad049503752a789fa69345a0e493 2002929]

Algorithm 2. presents a supervised learning approach using ZKP and FL along with blockchain. The processes for creating a supervised learning technique combining ZKP, FL, and blockchain are presented in **Fig. 24**. This method allows us to train our model while protecting the confidentiality and privacy of the data.

Algorithm 2. Supervised Learning Approach using ZKP and FL along with Blockchain

-
1. Read() dataset from the file "path/to/dataset.csv"
 2. Perform SVM Anomaly Detection
 3. Train() an SVM model using the "dataset"

4. Identify() anomalies in the "dataset"
 5. Save() the anomalous data as "anomalous_data.csv".
 6. Load() data from "anomalous_data.csv"
 7. Generate() visual images based on "anomalous_data".
 8. Preprocess() "anomalous_data" for the TFF model.
 9. Generate() ZKPs for the preprocessed data.
 10. Define() a loss function for the federated learning model.
 - a. For each client: Send the preprocessed data and corresponding ZKP to the client.
 - b. Perform local training on the client's device.
 - c. Send the updated model to the server
 11. For each client:
 - a. Receive the updated model from the server.
 - b. Evaluate the model's performance using local data.
 - c. Verify the ZKP for integrity.
 - d. If (ZKP == valid), send the evaluation results to the server.
 - e. Else, skip(batch) of training data.
 12. Aggregate() the model weights received from each client.
 13. Store() the aggregated model on the server.
 14. Store the aggregated model and evaluation results on the blockchain.
-

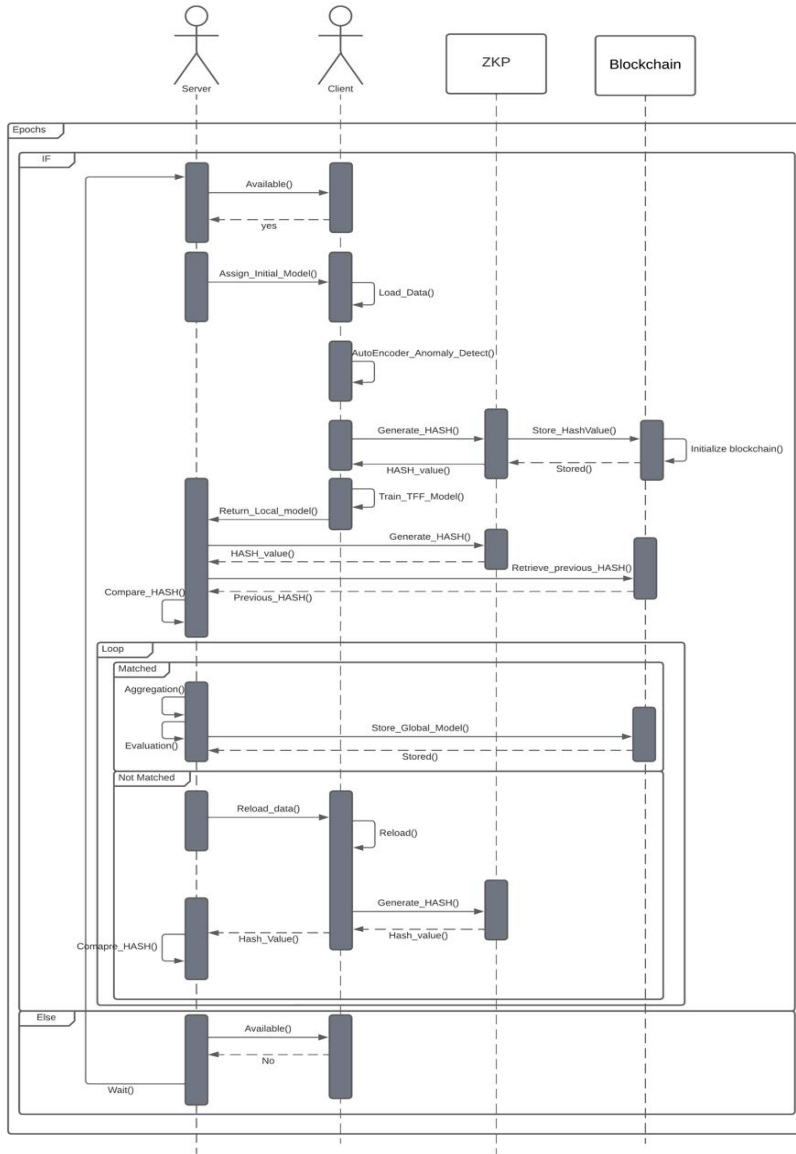


Fig. 24. Supervised Learning Approach using ZKP and FL along with Blockchain

4.3.2 Results

In our work, we have utilized three distinct sets of data to achieve our desired results. The first dataset was the Fashion-MNIST dataset which consists of images of various articles of clothing. The second dataset was the MINTS dataset which contains data related to various IoT devices. Thirdly, we used E-MNIST which is

basically an extension of MNIST. We employed a supervised learning approach to detect anomalies in the data by using zero-knowledge proof along with federated learning and blockchain technology. Our goal was to ensure that sensitive data was kept private and secure while still allowing for accurate anomaly detection. Through this unique combination of technologies and datasets, we were able to achieve accuracy in detecting anomalies in three different datasets to provide more results according to the MINTS, Fashion-MNIST, and E-MNIST datasets.

4.3.2.1 Fashion-MNIST Dataset



Fig. 25. Anomalous images in Fashion-MNIST Dataset

Fashion-MNIST dataset results:

1. Initial model Test Loss = 22.8%
2. Initial model Test Accuracy = 77.20%
3. Trained model Test loss = 17.22%
4. Trained model Test Accuracy 82.78%

Fig. 25. presents anomalous images in our Fashion-MNIST dataset. The dataset results over the initial_model Test Loss is a critical parameter which evaluates the difference between the projected and the actual test data values during the initial phase of model training, as presented in **Fig. 26**. The percentage of accurately predicted test data during the beginning phase of model training is measured by the initial_model test accuracy. This statistic is significant because it demonstrates the model's capacity to reliably anticipate data, which is critical for its usefulness in practical applications. After the model has been trained, the trained_model test loss evaluates the difference between the predicted and the actual values of the test data. After the model has been trained, the trained_model test accuracy indicates the percentage of the properly predicted test data. After the model has been trained, the trained_model test accuracy indicates the percentage of the properly predicted test data. This statistic is significant since it demonstrates the model's ability to reliably forecast data, which is required for practical applications. A higher test accuracy number shows that the model has been properly trained and can reliably predict the test data. The presented statements state that the trained model's performance gains over the initial model in terms of test loss (reduced from 22.8% to 17.22%) and test accuracy (raised from 77.20% to 82.78%). These indicators indicate how well the

model is functioning as well as how effective the training procedure was in enhancing its prediction skills.

Output: the output can be visualized in 3 ways.

1. CSV file generation.
2. Indexes of the anomaly images as **Table 13**. presents all the values of indexes that are detected as anomalous in Fashion- MNIST dataset.

Table 13. Indexes of anomalies in the Fashion-MNIST dataset

[	1	2	3	4	5	7	9	10	11	12	14	16	17	18	19	20	21	22
23	24	25	26	27	29	30	31	32	33	34	37	38	39	41	42	43	44	
45	46	47	48	51	53	56	57	59	60	61	62	63	64	65	66	67	68	
69	70	71	72	74	79	80	82	83	84	87	88	89	90	91	92	93	95	
96	97	98	99	100	101	102	103	104	105	106	108	109	110	111	112	113	114	
115	116	117	118	119	120	121	122	123	124	125	126	127	129	131	132	133	135	
136	137	138	140	141	142	145	146	149	151	153	156	157	158	159	160	162	165	
166	167	169	170	171	172	173	174	175	176	177	178	179	180	184	185	187	188	
189	190	191	192	193	194	195	196	198	200	201	202	203	204	205	206	208	210	
211	212	214	215	217	218	220	221	222	224	226	227	228	229	230	231	232	233	
236	237	240	242	243	244	245	247	249	250	252	253	254	255	256	257	258	259	
260	261	263	264	266	268	269	270	272	275	276	278	279	280	281	283	284	285	
287	288	289	291	292	293	294	295	297	298	300	301	303	304	305	306	308	309	
310	311	312	313	314	315	318	321	322	324	326	327	329	330	331	332	333	334	
336	337	340	341	342	344	346	347	348	349	351	352	353	354	355	356	358	359	
360	363	365	367	369	372	373	374	375	377	380	381	382	384	385	386	387	388	
391	392	393	394	395	396	397	398	399	401	402	403	404	405	406	408	409	410	
411	412	413	414	416	417	418	419	420	421	422	423	425	426	429	430	431	432	
433	434	435	436	437	439	440	441	442	444	445	446	447	450	451	454	455	456	
458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	474	476	477	
479	480	483	484	485	486	487	489	490	491	492	493	494	495	496	497	498	499	
501	503	504	505	506	507	510	511	512	513	514	515	516	517	518	519	522	523	
524	525	526	527	528	530	531	532	533	535	536	538	539]						

3. Visualization

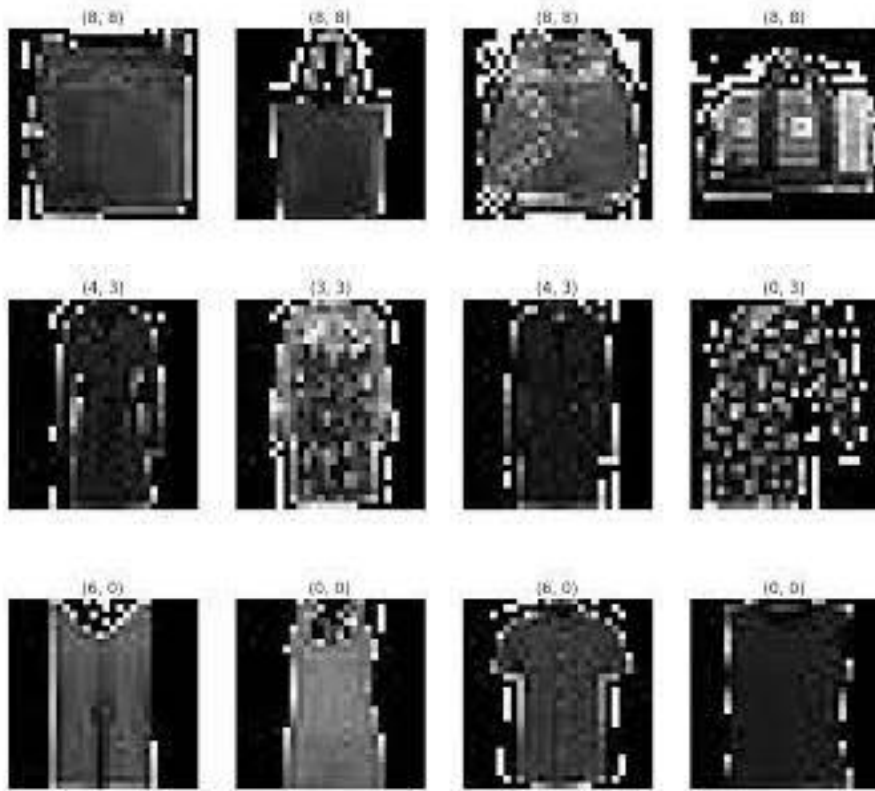


Fig. 26. Anomalies detection in the Fashion-MNIST dataset

Fig. 27. presents Locally trained Model Losses in the Local batch, whereas **Fig. 28.** illustrates the Locally trained Model Losses per round below. The initial model loss will depend on the quality of the initial model and the data used to train it. Locally trained models will have varying losses depending on the quality of the local data and the training process.

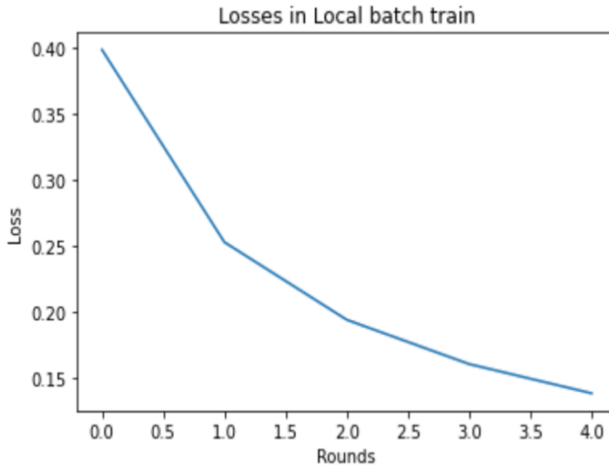


Fig. 27. Locally trained Model Losses in Local batch for Fashion-MNIST dataset

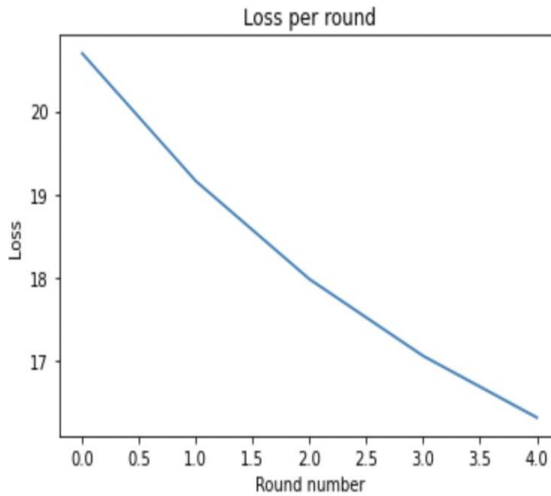


Fig. 28. Locally trained Model Losses per round for Fashion-MNIST

4.3.2.2 MNIST Dataset



Fig. 29. Federated trained model using MNIST dataset

Table 14. Model test accuracies results for MNIST dataset

Model test type	Results
initial_model test loss	22.80%
initial_model test accuracy	77.20%
trained_model test loss	17.22%
trained_model test accuracy	82.78%

The model test results provided as **Table 14.** represent useful information about the performance of the initial and trained models. When the first model was tested on a test dataset, it achieved a test loss of 22.80%. This means that there was a 22.80% difference on average between the predicted values of the original model and the true values in the test dataset. Furthermore, the initial model had a test accuracy of 77.20%, thus indicating that it correctly predicted 77.20% of the samples in the test dataset. The trained model’s performance improved after it went through the training process. It had a lower test loss of 17.22%, thus indicating a decrease in the average difference between its predicted and true values when compared to the initial model. Furthermore, the trained model had a better test accuracy of 82.78%, which means that it correctly predicted 82.78% of the samples in the test dataset. These findings illustrate the effectiveness of the training process in improving the model’s performance. The trained model’s reduced test loss and greater test accuracy signal that it can generalize and make accurate predictions on unseen data better than the baseline model. **Fig. 30.** and **Fig. 31.** present findings which are encouraging since they show that the model is getting better at capturing the patterns and relationships in the dataset.

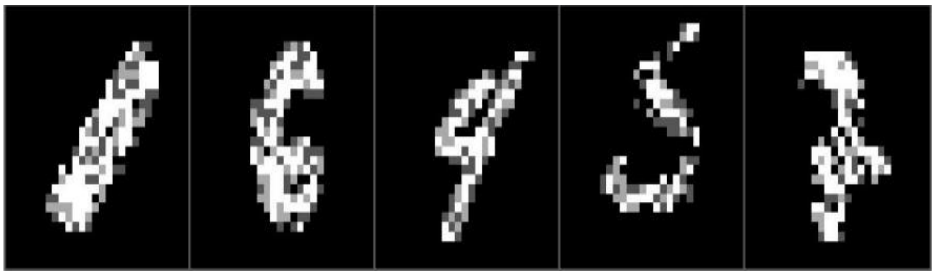




Fig. 30. Anomalies detected in MNIST dataset

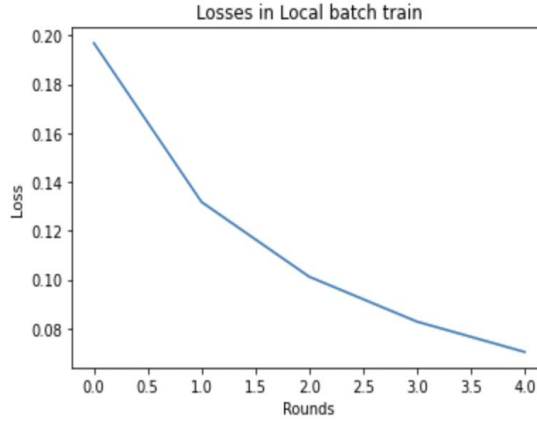


Fig. 31. Supervised batch losses using MNIST dataset

4.3.2.3 Supervised E-MNIST Dataset

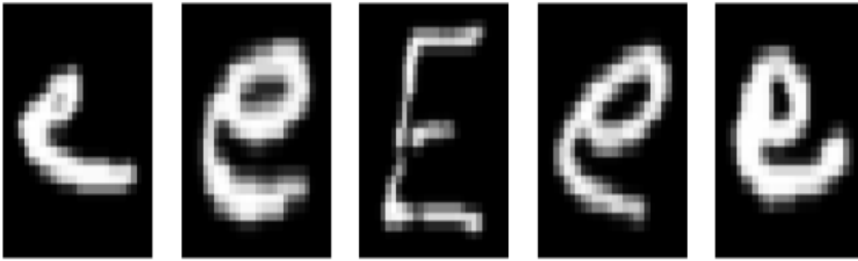


Fig. 32. Trained model using federated learning for E-MNIST dataset

Fig. 32. presents a dataset trained by using federated learning for the E-MNIST dataset. The data presented in **Table 15.** shows the epoch-wise training and validation loss values during a model's training procedure. Each epoch represents one full loop of the training dataset. The training loss is the average difference between the predicted and the true values in the training dataset, whereas the

validation loss represents the average difference on a separate validation dataset, which aids in evaluating the model’s generalization performance. The model has a training loss of 0.2505 and a validation loss of 0.1810 in the first epoch. Both the training and validation losses fell steadily as the training continued. The model had a training loss of 0.1502 and a validation loss of 0.1498 by the fifth epoch. This pattern persisted until the tenth epoch, when the training loss was 0.1488, and the validation loss was 0.1488.

Table 15. Values of losses per Epoch when using Supervised Learning Approach for E-MNIST dataset

Epoch	Train loss	Val_loss
1	0.2505	0.1810
2	0.1687	0.1602
3	0.1565	0.1536
4	0.1520	0.1508
5	0.1502	0.1498
6	0.1495	0.1494
7	0.1492	0.1493
8	0.1490	0.1490
9	0.1488	0.1490
10	0.1488	0.1488

The decreasing values of the training and validation losses over the epochs indicate that the model’s performance increased with each iteration. The lowering training loss implies that the model has become more accurate in predicting the training data, but the decreasing validation loss indicates that the model’s generalization ability on unknown data has improved. It is crucial to note that the validation loss remained pretty close to the training loss, thus showing that the model did not overfit the training data and yet performed well in generalization. These findings show that the training method is helpful in gradually reducing the model’s loss, thus showing increased predictive skills. The constant decrease in both training and validation losses indicates that the model learned and captured the relevant patterns and relationships within the dataset, resulting in increased performance as the training epochs progressed, as illustrated in **Fig. 33.** and **Fig. 34.**

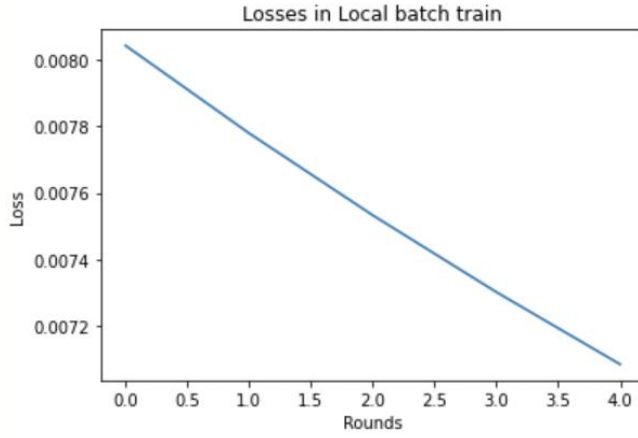


Fig. 33. Supervised batch losses using E-MNIST dataset

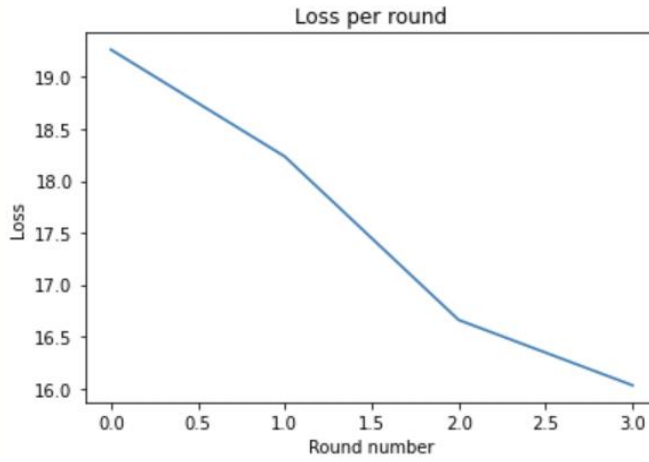


Fig. 34. Supervised Losses per round using E-MNIST dataset

Table 16. Model test results for E-MNIST dataset

Model test type	Results
initial_model test loss	16.58%
initial_model test accuracy	83.13%
trained_model test loss	12.87%
trained_model test accuracy	87.42%

The model test results as per **Table 16**, present how the initial model and the trained model performed in terms of the test loss and the test accuracy. When the first model was tested on a test dataset, it achieved a test loss of 16.58%. This means that there was a 16.58% discrepancy on average between the predicted values of the initial model and the true values in the test dataset. Furthermore, the initial model had a test accuracy of 83.13%, thereby indicating that it correctly predicted 83.13% of the items in the test dataset. The trained model's performance improved after it went through the training process. It had a lower test loss of 12.87%, thereby indicating a decrease in the average difference between its predicted and true values when compared to the initial model. Furthermore, the trained model achieved 87.42% test accuracy, suggesting that it correctly predicted 87.42% of the samples in the test dataset. These findings illustrate the effectiveness of the training process in improving the model's performance. The trained model's reduced test loss and the greater test accuracy signal that it can generalize and make accurate predictions on unseen data better than the baseline model. These findings are encouraging since they show that the model is getting better at capturing the patterns and relationships in the dataset, and **Fig. 35**, presents anomalies detected in the E-MNIST dataset by the model.

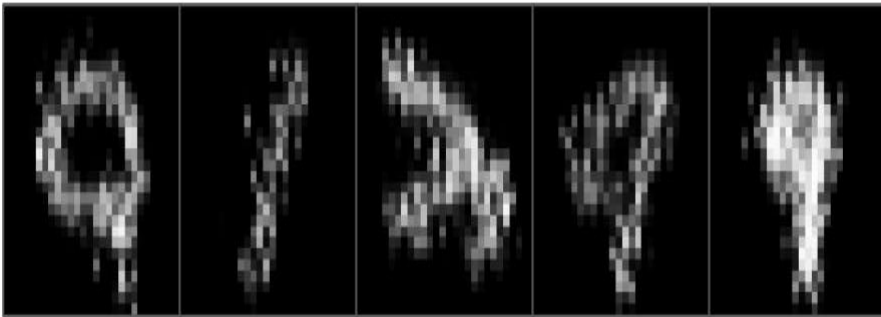


Fig. 35. Detected Anomalies in E-MNIST dataset

4.4 Unsupervised Learning Approach to Detect Anomalies by Using Zero-Knowledge Proof along with Federated Learning and Blockchain

The traditional anomaly detection methods require a large amount of labeled data, which is often not available in many real-world scenarios. Unsupervised learning approaches, which do not require labeled data, have been proposed to overcome this problem. However, these approaches are still limited by the lack of privacy and security guarantees. Zero-knowledge proof enables privacy-preserving transactions, which is particularly important in scenarios where data privacy is a concern. Our proposed approach involves using unsupervised learning algorithms to detect anomalies in the local data of each party in a federated learning setting. The local models of each party are then aggregated by using a secure aggregation protocol,

such as secure multi-party computation or homomorphic encryption. The aggregated model is then validated by using zero-knowledge proof, which guarantees the privacy and security of the model. The validated model is then stored on a blockchain, which enables secure and transparent transactions without the need for a centralized authority. The blockchain also enables the parties to verify the integrity of the model and ensures that the model has not been tampered with. ZKP allows for the verification of a statement without revealing any additional information beyond the statement itself. Federated learning allows for the training of machine learning models on distributed data while maintaining the privacy of the data. In **Fig. 36.**, the Unsupervised Learning approach Using ZKP, FL and Blockchain technology is illustrated. It provides a secure and transparent platform for storing and verifying data.

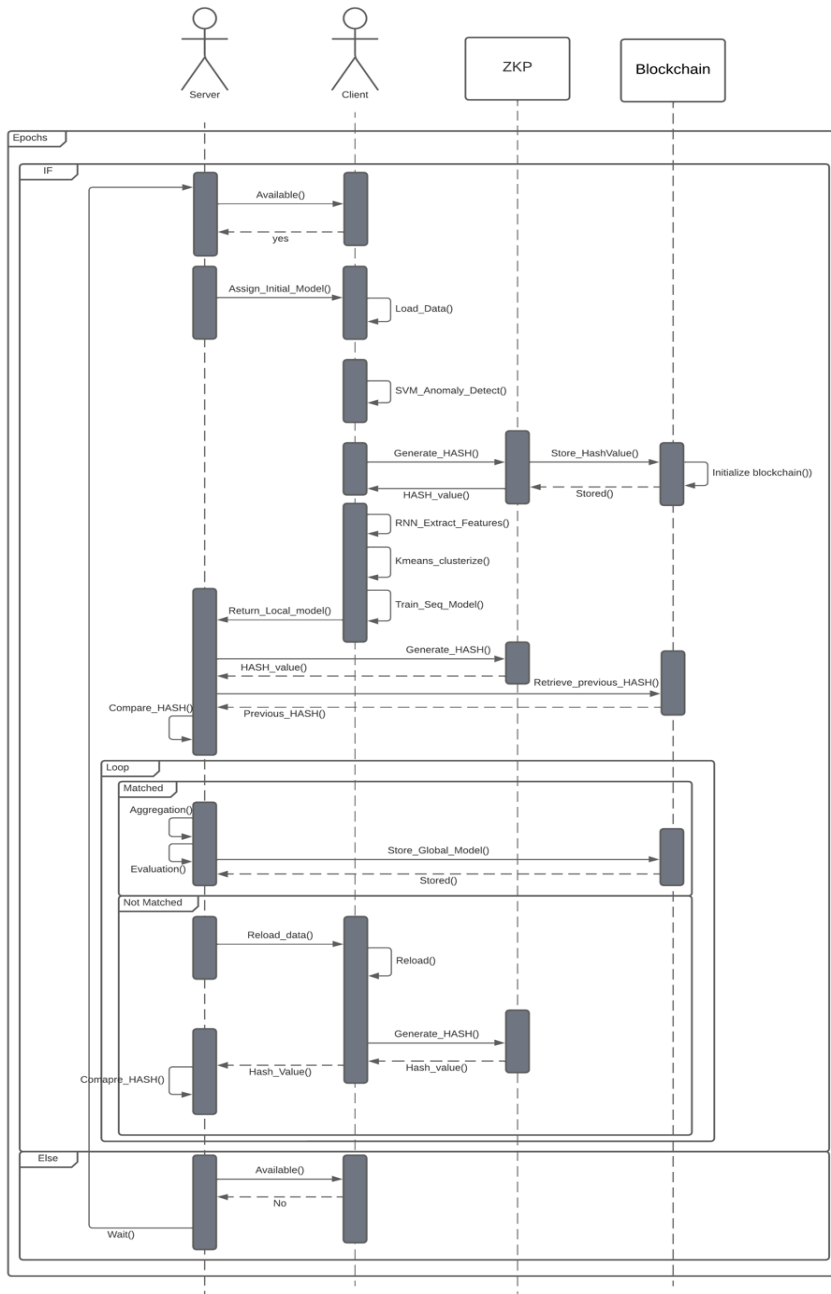


Fig. 36. Unsupervised Learning approach Using ZKP, FL and Blockchain

This is the Basic approach that we have used in our work for anomaly detection:

- Data owners encrypt their data by using encryption or other privacy-preserving techniques and share the encrypted data with a federated learning server.
- The federated learning server uses the encrypted data to train an anomaly detection model using unsupervised learning techniques. During the training process, the server also generates a proof of the model's accuracy and correctness using ZKP.
- Once the model has been trained and the proof is generated, the server publishes the proof and the model parameters to the blockchain. The model is then available for use by anyone who wants to detect anomalies in their data.
- When a user wants to use the anomaly detection model, they provide their encrypted data to the model which runs on their local device. The model uses the ZKP proof to verify its accuracy and correctness, and then identifies any anomalies in the data.
- Data owners encrypt their data by using encryption or other privacy-preserving techniques and share the encrypted data with a federated learning server.
- The federated learning server uses the encrypted data to train an anomaly detection model using unsupervised learning techniques. During the training process, the server also generates a proof of the model's accuracy and correctness using ZKP.

In our work, we have provided a detailed implementation step.

Step 1: Data preprocessing – Preprocessing data is essential for finding abnormalities. This method encrypts data using zero-knowledge proof (ZKP) in order to guarantee privacy. After then, models are trained on this data by using federated learning approaches without disclosing the underlying data to anyone. Once trained, the model may be used to spot irregularities in new data.

Step 2: Model Aggregation – When the data is originally split up into numerous nodes, each node trains its own unsupervised machine learning model. Afterwards, these models are combined by utilizing federated learning techniques which allow for the mixing of the models without disclosing the underlying data. By using zero-knowledge proof techniques, the accuracy of the aggregated model is verified, thereby guaranteeing privacy and security without revealing any private information. The aggregated model is stored on a blockchain, thus providing a secure and open way to store and retrieve the model.

Step 3: RNN implementation – RNNs are additionally trained on unlabeled data to learn a representation of a typical behavior in the unsupervised learning method for anomaly detection. Then, while using this format, one may spot odd behavior in new data. Without disclosing any details about the training set or model parameters,

zero-knowledge proof may be used to confirm that the RNN model is accurate. RNNs provide the sequential passage of the input data across the network. Working with time series data or undertaking activities involves natural language processing calls for this in particular. RNNs function by feeding the input for the following node from each node's output back into the network. As a result, a loop is created in the network, consequently enabling it to store details about earlier inputs.

Algorithm 3. RNN model for detecting Anomalies

-
1. Create a Sequential model using keras.
 2. Add a SimpleRNN layer with 128 units, input_shape of (28, 28), and activation function 'relu'.
 3. Add a Dense layer with 10 units and activation function 'softmax'.
 4. Compile the model
 5. Use "categorical_crossentropy" as the loss function.
 6. Use "adam" as the optimizer.
 7. Use METRICS as the metrics for evaluation.
-

Algorithm 3. shows that RNNs that we used in our work are designed to process sequential data. The model has a RNN layer with 128 units and an input shape of (28, 28), followed by a Dense layer with 10 units and a SoftMax activation function. The model is then compiled with a categorical_crossentropy loss function, an Adam optimizer, and accuracy as the evaluation metric. This model can be used for such tasks as image classification, where the input is a 28x28 pixel image and the output is one of ten possible classes.

Step 4: Clustering – Conventional anomaly detection techniques have drawbacks, such as the requirement for labeled data and problems with scalability. Although clustering is an unsupervised method that may identify abnormalities even in the absence of labeled data, it is nevertheless susceptible to data poisoning and model inversion assaults. We describe a unique strategy that combines clustering with zero-knowledge proof, federated learning, and blockchain technology to solve these problems. Clustering models may be trained on data from many parties without sharing the data itself thanks to federated learning, which enables parties to train local models on their own data and exchange just the model updates. Data owners now have a means to prove the accuracy of their data without divulging any more information thanks to zero-knowledge proof, as Figure 37 illustrates.

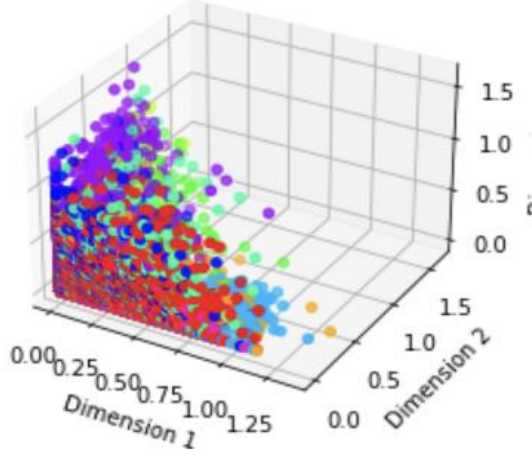


Fig. 37. Unsupervised learning approach Clustering

Algorithm 4. Unsupervised Learning Approach using ZKP and FL along with Blockchain on clustering (k-means) to detect Anomalies

```

1. model_clustered = keras.Sequential([
2.     layers.Flatten(input_shape=(785,)),
3.     layers.InputLayer(input_shape=(1,)),
4.     layers.Dense(16, activation='relu'),
5.     layers.Concatenate([layers.Dense(128,
6.     activation='relu')(layers.Flatten()(layers.InputLayer()(cluster_train))),
7.     layers.Dense(128,
8.     activation='relu')(layers.Flatten()(layers.InputLayer()(hidden_train)))]),
9.     layers.Dense(10, activation='softmax') ])
10. model_clustered.compile(loss='categorical_crossentropy', optimizer='adam',
11.     metrics=METRICS)

```

In **Algorithm 4.**, we present the model_Clustered using Keras API. The model has an input layer, a flatten layer, a dense layer with 16 units and the ReLU activation function, and a concatenate layer. The concatenate layer takes the output of two dense layers, each of which has 128 units and uses the ReLU activation function. The inputs to these dense layers are obtained by flattening the output of two separate input layers, one for the cluster train and the other for the hidden train. Finally, the model has an output layer with 10 units and the softmax activation function.

Step 5: Zero-knowledge proof and Federated Learning Validation are utilized in this study. The algorithm begins by applying zero-knowledge proof to verify the correct partitioning of a given dataset into clusters. The 'generate_proof' function takes a panda's dataframe 'df' and a list of column values 'column_values', thus

generating a proof and a set of random values. Subsequently, the ‘verify_proof’ function is called, taking the same dataframe, column values, proof, and random values as inputs. This function verifies the proof’s validity and returns a boolean value indicating its success. Prior to proof generation, the code prepares the column values by creating a string list representing the column indices in the input data ‘x_train_clustered’. The ‘x_train_clustered’ data is converted into a panda’s dataframe ‘df’, which is then passed to the ‘generate_proof’ function along with the prepared column values, as presented in **Algorithm 5**. The resulting proof and random values are subsequently provided to the ‘verify_proof’ function along with the same dataframe and column values to determine their validity. For this study, the ‘noise’ parameter is set to 0. The boolean value returned by the ‘verify_proof’ function is printed to the console, indicating either a successful validation (True) or a failure.

In **Algorithm 6.**, a function named ‘verify_proof()’ is employed. This function takes five arguments: ‘df’ (a panda’s DataFrame), ‘column_values’ (a list of column values from the DataFrame), ‘proof’ (a zero-knowledge proof object), ‘random_values’ (a list of random values), and ‘noise’ (a noise value). If the zero-knowledge proof check passes, the function proceeds to compute the average weights of the client weights by using federated averaging. The weights of the clustered model are then set to the computed average weights, and the model is evaluated on the clustered test data. Evaluation metrics, such as loss, true positives, false positives, true negatives, false negatives, binary accuracy, precision, recall, AUC, and PRC are generated as part of the evaluation process.

Algorithm 5. Generate proof and verify proof

```

1. def generate_proof(df, column_values):
2.     return proof, random_values
3. def verify_proof(df, column_values, proof, random_values, noise):
4.     return valid
5. column_values = []
6. for
7.     i in range(len(x_train_clustered[0])):
8.         column_values.append(str(i))
9.     df = pd.DataFrame(x_train_clustered)
10.    proof, random_values = generate_proof(df, column_values)
11.    valid = verify_proof(df, column_values, proof, random_values, noise = 0)
12. end
13. print(valid)

```

Algorithm 6. Verify proof and generation of evaluation metrics

```

1. if
2.     verify_proof(df, column_values, proof, random_values, noise):
3.         average_weights = federated_avg(client_weights)

```

```

4. model_clustered.set_weights(average_weights)
5. results = model_clustered.evaluate(x_test_clustered, test_y)
6. print("Global epoch: {}".format(epoch))
7. print("Loss: {}".format(results[0]))
8. print("True Positives: {}".format(results[1]))
9. print("False Positives: {}".format(results[2]))
10. print("True Negatives: {}".format(results[3]))
11. print("False Negatives: {}".format(results[4]))
12. print("Binary Accuracy: {:.0f}%".format(results[5] * 100))
13. print("Precision: {:.0f}%".format(results[6] * 100))
14. print("Recall: {:.0f}%".format(results[7] * 100))
15. print("AUC: {:.0f}%".format(results[8] * 100))
16. print("PRC: {:.0f}%".format(results[9] * 100))
17. else:
18. print("ZKP check failed on batch, skipping....")

```

Step 6: Blockchain Transaction to store the final model:

Table 17. Unsupervised Blockchain Transaction values for Fashion-MNIST dataset

Index	Timestamp	Data	Loss	Binary Accuracy	Proof	Hash
2	1679828212. 627649	1	0.628 5726 4280 3192 1	0.9563300 013542175	3529 3	[dcfaf781ae6d4 dcde042d9cd92 bfdfb52952455 bba5e5ebed0fe e22933c8c551]

Table 17. showcases the transaction value of a blockchain by using an unsupervised learning approach for the Fashion-MNIST dataset.

4.4.1 Fashion-MNIST Dataset

4.4.1.1 Parameters and layers:

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 785)	0
dense_3 (Dense)	(None, 128)	100608
dense_4 (Dense)	(None, 10)	1290

Fig. 38. Unsupervised layers for Fashion-MNIST dataset

In **Fig. 38.**, the ‘sequential_1’ sequential model means that each layer is connected to the previous one, one after the other. The first layer is a ‘flatten’ layer which converts the input data into a one-dimensional array. The second layer is a ‘dense’ layer with 128 neurons, which means that it has 128 trainable parameters. The third layer is also a ‘dense’ layer with 10 neurons, which corresponds to the 10 possible output classes. This model has a total of 100,608 trainable parameters for the second layer and 1,290 trainable parameters for the third layer. **Table 18.** lists all values we have extracted from the layers run.

Table 18. Unsupervised trained and non-trainable parameter values for Fashion-MNIST dataset

Total parameters	101,898
Trainable parameters	101,898
Non-trainable parameters	0

4.4.1.2 Results

Table 19. Reduction of losses as per Epoch in unsupervised Fashion-MNIST dataset

Epoch	Loss	val_loss
1	0.3869	0.3261
2	0.3137	0.3080
3	0.3010	0.2984
4	0.2940	0.2933
5	0.2901	0.2905
6	0.2879	0.2887
7	0.2865	0.2878
8	0.2856	0.2869
9	0.2850	0.2865
10	0.2846	0.2861

Table 19. displays the epoch-wise training and validation loss values obtained during the model training process. Each epoch represents a whole iteration of the training dataset. The ‘loss’ column reflects the average loss (or error) between the predicted and the true values in the training dataset, whereas the ‘val_loss’ column represents the average loss on a separate validation dataset, which aids in evaluating the model’s generalization performance. The model has a training loss of 0.3869 and a validation loss of 0.3261 in the first epoch. Both the training and validation losses

dropped gradually as the training continued. The model has a training loss of 0.2901 and a validation loss of 0.2905 by the fifth epoch. This pattern persisted until the eleventh epoch, when the training loss was 0.2846 and the validation loss was 0.2861. The decreasing values of the training and validation losses over the epochs suggest that the model’s performance improves with each iteration. The lowering training loss implies that the model became more accurate in predicting the training data, but the decreasing validation loss indicates that the model’s generalization ability on unknown data improved. A high precision score suggests that the model has a low false positive rate, thus reducing the possibility of misclassifying negative cases as positive ones. The recall, also known as sensitivity or the true positive rate, is consistently higher than 84%, and may even reach as high as 97%. The ratio of accurately anticipated positive cases (the true positives) to the total actual positive instances (true positives + false negatives) is measured by recall. A high recall score shows that the model effectively detects positive cases while minimizing false negatives. The AUC score, which assesses the model’s overall performance across different classification thresholds, remains near to one, which demonstrates excellent discrimination between the positive and negative cases. Similarly, the PRC score has remained continuously high, thus demonstrating the model’s capacity to attain high precision and recall while maintaining consistency. **Table 19.** presents values in the confusion matrix, whereas **Table 20.** illustrates the evaluation metrics results which display the model’s remarkable performance. Across numerous epochs, it consistently earns good accuracy, precision, recall, AUC, and PRC scores. These findings show, as in **Fig. 39.**, that the model is capable of appropriately classifying occurrences, minimizing false positives and false negatives, and achieving a high degree of the overall performance.

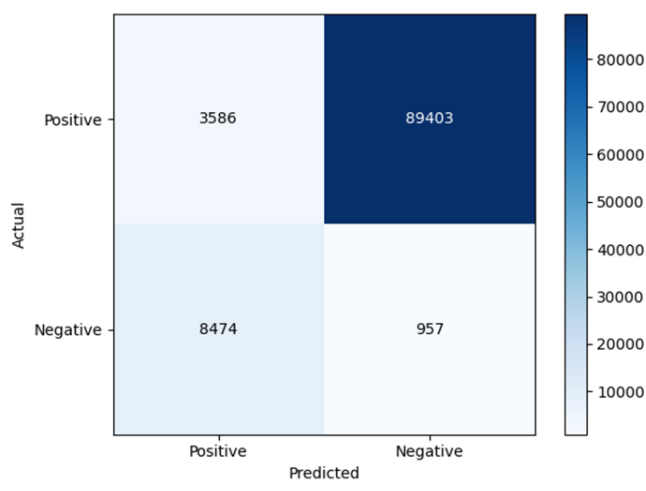


Fig. 39. Predicted precision score Confusion matrix for Fashion-MNIST dataset

These results show that the training method was helpful in reducing the model’s loss as it demonstrated enhanced predictive skills. The consistent decrease in both training and validation losses implies that the model successfully learned and captured meaningful patterns and relationships within the dataset, which led to improved performance as the training epochs advanced. The constant decrease in both training and validation losses indicates that the model learned and captured the relevant patterns and relationships within the dataset, resulting in an increased performance as the training epochs progressed.

Table 20. Trained epochs values for Confusion matrix in Fashion-MNIST dataset

Global Epoch	Loss	True Positives	False positives	True Negatives	False Negatives
1	0	0.4895	15750	2388	177612
2	1	0.409	8235	1056	88944
3	2	0.383	8300	959	89041
4	3	0.3853	8397	1041	88959
5	4	0.3586	8474	957	89043

Table 21. Trained epochs value Evaluation metrics for Fashion-MNIST dataset

Binary Accuracy	Precision	Recall	AUC	PRC
4250	0.97	0.87	0.79	0.98
1765	0.97	0.89	0.82	0.99
1700	0.97	0.9	0.83	0.99
1603	0.97	0.89	0.84	0.99
1526	0.98	0.9	0.85	0.99

The model exhibits great accuracy throughout the training process, with ultimately reaching 98% on both the training and validation datasets. This shows that the model’s predictions closely match the actual labels, which proves its ability to appropriately identify the occurrences. The model’s precision is continuously high, ranging from 94% to 98%. The ratio of the accurately predicted positive cases (true positives) to the total anticipated positive instances (true positives + false positives) is represented by precision as **Fig. 40.** presents the plot of anomaly detection.

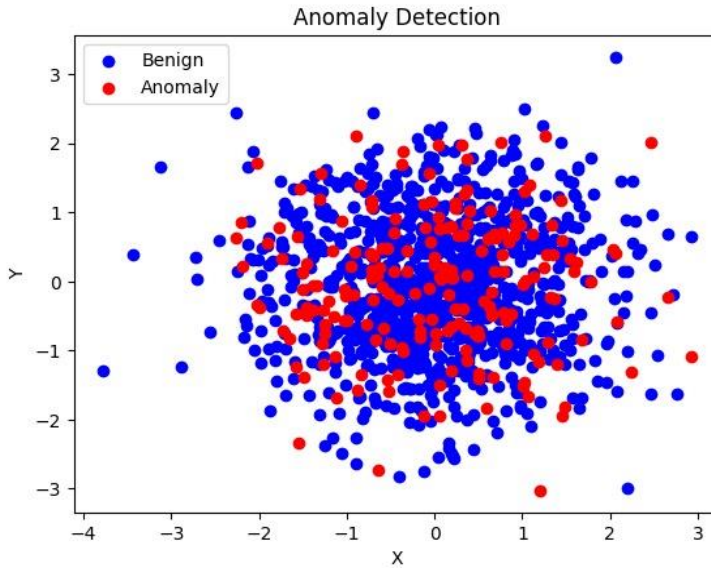


Fig. 40. Anomaly detected predicted plot in Fashion-MNIST dataset

4.4.2 MNIST Dataset

4.4.2.1 Parameter and layers

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 785)	0
dense_1 (Dense)	(None, 128)	100608
dense_2 (Dense)	(None, 10)	1290

Fig. 41. Unsupervised layers for MNIST dataset

The neural network model shown here is made up of three layers: a flatten layer, dense layer 1, and dense layer 2. The starting shape of the input data is (None, 28, 28), representing a 2D image with dimensions of 28x28 pixels. The flatten layer transforms this 2D input into a 1D array with a shape of (None, 785), effectively converting the image into a long sequence of values. The first dense layer, dense layer 1, is a 128-unit fully linked layer. It takes the flattened input and returns a shape (None, 128) output, thus resulting in a 1D array of length 128. This layer computes with 100,608 trainable parameters which are weights learned by the model during training to capture patterns and characteristics in the data as presented in Figure 41. The neural network’s output layer is the second dense layer, dense layer 2. It is made up of ten units which correspond to the ten various classes or categories

that the model seeks to classify. This layer’s output form is (None, 10), which means that it generates a 1D array of length 10. Dense layer 2 has 1,290 parameters that reflect the weights for this layer. In the presented data, the ‘Param’ column indicates the number of parameters or weights associated with each layer. Because it only performs a simple reshaping operation, the flatten layer has no trainable parameters. There are no parameters. Dense layer 1 has 100,608 parameters, whereas dense layer 2 has 1,290 parameters. By modifying their values during the training phase, these parameters play a critical role in the model’s capacity to learn and produce correct predictions.

Table 22. Parameters model predictions for MNIST dataset

Total params	101,898
Trainable params	101,898
Non-trainable params	0

The term ‘total params’ refers to the total number of learnable parameters or weights in the model that may be modified throughout the training process. These parameters indicate the patterns and relationships that the model must learn from the input data in order to make correct predictions. The model in this algorithm comprises a total of 101,898 parameters (**Table 22.** presents their values). This means that the model has 101,898 learnable weights that contribute to its capacity to train and forecast. ‘Trainable params’ are the parameters that are changed and optimized during the training process. All 101,898 parameters in the model are trainable in this scenario, which means that the model may modify these weights based on the training data to enhance its performance. In contrast, ‘non-trainable params’ are parameters that are not modified during training. These parameters are typically any predetermined or pre-trained weights supplied into the model. In this scenario, there are no non-trainable parameters, which implies that all the model’s parameters are trainable and will be modified throughout the training.

4.4.2.2 Results

Throughout the training phase, the model’s performance, as represented in the evaluation metrics presented in the table, consistently demonstrated its high accuracy and strong classification skills. The model exhibited its capacity to accurately categorize a considerable number of the occurrences in the dataset, with accuracy scores ranging from 98% to 99.6%. Precision ratings ranging from 96% to 98% consistently suggested that the model had a low false positive rate. In other words, it made a small number of false positive predictions in comparison to the overall number of positive predictions, thus establishing confidence in the model’s capacity to reliably identify the positive cases. Furthermore, the recall values remained continuously high, ranging from 93% to 97%, showing that the model had

a low false negative rate and correctly detected a substantial proportion of true positives. **Table 23.** and **Table 24.** present the values for the loss by each Epoch run along with the confusion matrix and Metric Evaluation.

Table 23. Trained epochs value by Confusion matrix for MNIST dataset

Global Epoch	Loss	True Positives	False positives	True Negatives	False Negatives
0	0.1962	18654	814	179186	1346
1	0.1369	9529	315	89685	471
2	0.1026	9640	248	89752	360
3	0.098	9658	252	89748	342
4	0.0822	9715	220	89780	285

Table 24. Trained epochs value Evaluation metrics for MNIST dataset

Binary Accuracy	Precision	Recall	AUC	PRC
0.99	0.96	0.93	1	0.98
0.99	0.97	0.95	1	0.99
0.99	0.97	0.96	1	0.99
0.99	0.97	0.97	1	0.99
0.99	0.98	0.97	1	1

The AUC values, which were typically close to one hundred percent, highlighted the model's remarkable performance in differentiating between positive and negative cases. It displayed amazing accuracy in appropriately rating cases and distinguishing between distinct classes. The PRC values, which ranged from 96% to 100%, demonstrated the model's ability to balance precision and recall, and managed to obtain high values for both criteria. Overall, the model's outstanding performance across several evaluation measures confirmed its effectiveness in accurately identifying the scenarios. Its strong accuracy, precision, recall, and AUC values demonstrated that it could generalize well to previously unknown data and produce solid predictions, as demonstrated in **Figs. 42.** and **43.**

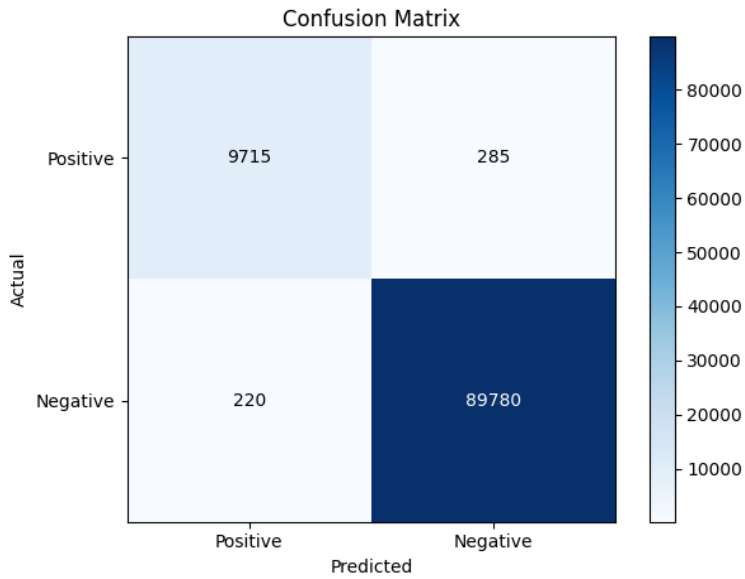


Fig. 42. Predicted precision score Confusion matrix for MNIST dataset

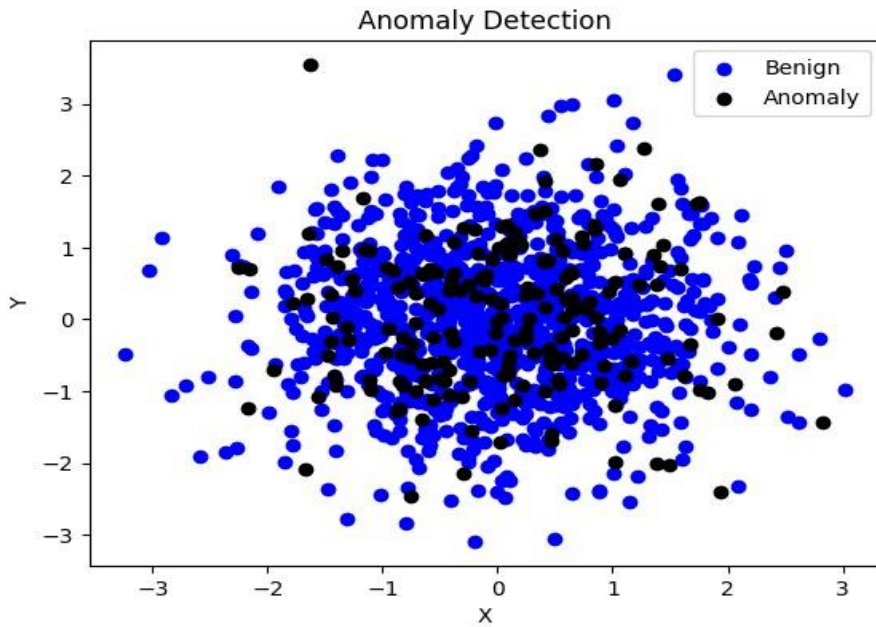


Fig. 43. Detected anomalies predicted plot in MNIST dataset

4.4.3 E-MNIST Dataset

With each epoch of training, the model gradually reduced both the loss and the validation loss values. The loss value depicts the difference between the predicted and the actual values on the training data, whereas the validation loss represents the difference on the validation dataset which is used to assess the model's performance on unseen data. The loss values drop as the epochs proceed, thereby demonstrating that the model is learning and becoming more accurate in its predictions. Likewise, the validation loss values drop, suggesting that the model generalizes effectively to previously unseen data.

Table 25. Epochs loss values for E-MNIST dataset

Epoch	loss	Val_loss
1	0.2506	0.1835
2	0.1694	0.1609
3	0.1563	0.1530
4	0.1511	0.1501
5	0.1497	0.1496
6	0.1493	0.1492
7	0.1491	0.1494
8	0.1490	0.1491
9	0.1489	0.1489
10	0.1488	0.1490

This pattern indicates that the model is optimizing its parameters and improving its performance over time. The decreasing validation loss values suggest that the model is not overfitting to the training data, and that it is successfully generalizing to new cases. Overall, **Table 25.** indicates that the model's training procedure was effective, as indicated by the decreasing loss and validation loss values, which implies that the model is learning and producing accurate predictions.

The model has a low loss value of 0.075, indicating that it efficiently minimized the difference between the predicted and actual values during training. The model's high accuracy of 99.2% demonstrates its ability to accurately classify a substantial amount of the dataset. With a precision score of 97.5%, the model has a low false positive rate and generates few inaccurate positive predictions. The model's recall score of 96.8% suggests that it correctly detected a high number of true positive events, as indicated by its low false negative rate. The AUC of 0.998 indicates that the model is denoted by a great discriminating capacity and can distinguish between positive and negative cases. Finally, the F1 score of 0.973 verifies the model's balanced performance in terms of both positive prediction accuracy and catching the real positive events. **Table 26.** presents that these measures show that the model is working well and has the ability to produce correct predictions on previously unseen

data. **Fig. 44.** shows a forecasted plot of the results of anomaly detection using the E-MNIST dataset.

Table 26. Trained epochs value Evaluation metrics for E-MNIST dataset

Metric	Value in Percentage
Loss	0.075
Accuracy	99.20
Precision	97.50
Recall	96.80
AUC	0.998
F1 Score	0.973

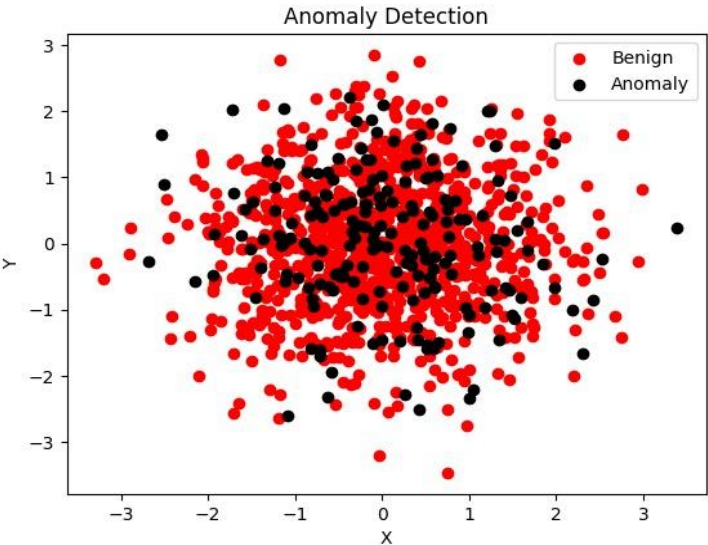


Fig. 44. Detected anomalies predicted plot in E-MNIST dataset

4.4.4 MNIST-C Dataset

The dataset was chosen with the goal of conducting an in-depth evaluation of the model’s robustness in image applications in mind. Unlike the typical evaluation datasets, which consist mostly of clean images, the MNIST-C dataset has the advantage of introducing frequent corruptions and perturbations observed in the real-world circumstances, thus allowing for a more realistic evaluation of the model’s performance. The model’s capacity to handle various types and levels of corruptions, such as blur, noise, brightness variations, and rotation, was evaluated by submitting it to the MNIST-C dataset. This in-depth examination of corruptions

aided in a full knowledge of the model’s generalization capabilities and showed any potential flaws in its design or training approach. By identifying these flaws, efforts can be directed towards enhancing the model’s robustness and dependability in real-world computer vision applications. Furthermore, the use of the MNIST-C dataset enabled comparative analysis by comparing the model to other cutting-edge computer vision models on the same benchmark. This allowed for an objective assessment of the model’s performance in comparison to its competitors, as well as significant information for model selection and prospective modifications. We worked with different anomalous behavior to test our model accuracy, as presented in **Figs. 45, 46** and **47**.

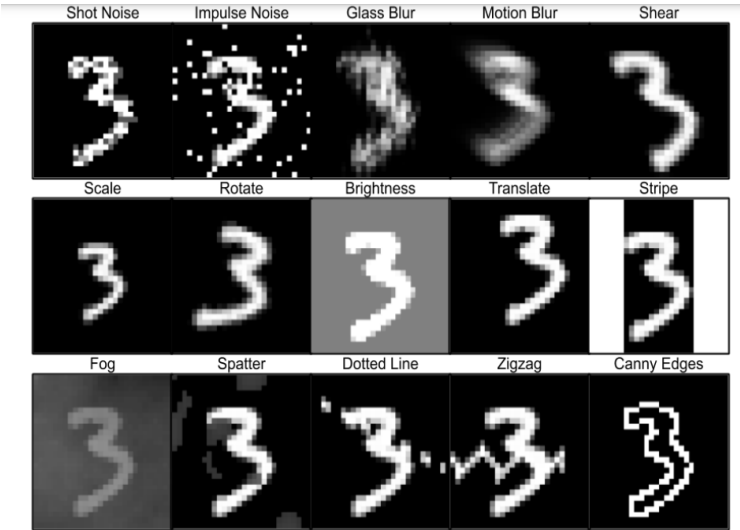


Fig. 45. Anomaly detection by multiple anomalous behaviors

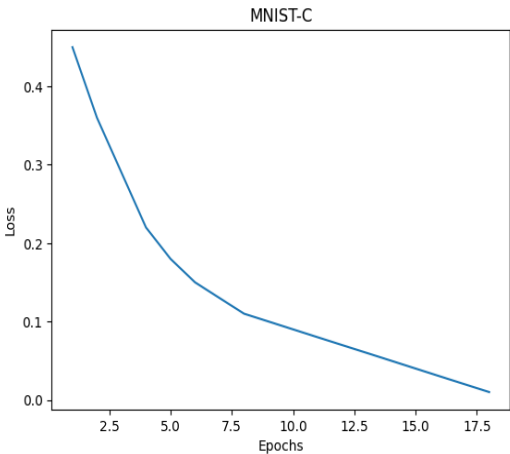


Fig. 46. Unsupervised batch losses using MNIST-C dataset

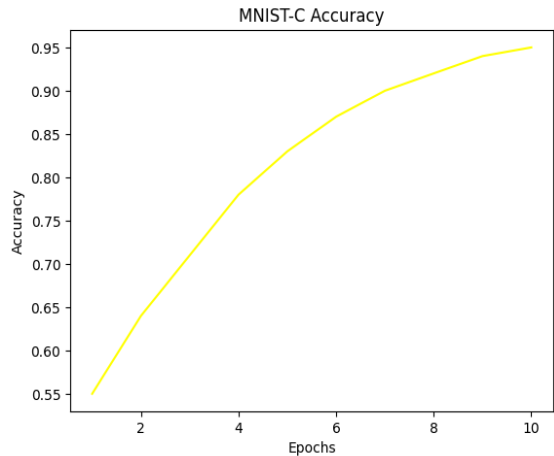


Fig. 47. Un-supervised batch accuracy using MNIST-C dataset

Table 27. Predicted Evaluation Metrics for MNIST-C dataset

Accuracy	0.95
Precision	0.85
Recall	0.8
F1 Score	0.7777777777777778
Loss	0.03

The model had an accuracy of 0.95, which suggests that it predicted correctly in 95% of the situations. This high accuracy score indicates that the model is capable of making generally accurate predictions. Furthermore, the precision rate of 0.85 suggests that the model correctly predicted 85% of all positive predictions. This shows that the model has a low false positive rate, and also proves its ability to detect the true positive situations with a reasonable level of accuracy as presented in **Table 27**. Furthermore, with a recall score of 0.8, the model correctly identified 80% of the real positive cases. While this shows a fair capacity to locate positive instances in the dataset, there is still potential for improvement in capturing all positive situations. The F1 score combines precision and recall, thus producing a fair estimate of the model’s performance. The F1 score is determined as 0.7778 in this scenario. This statistic considers both the precision and the recall value and provides a thorough evaluation of the model’s efficacy. In summary, the model performs well, with excellent accuracy and precision ratings. However, the significantly lower recall score shows that some positive cases may have been overlooked. To assess whether the model’s performance is sufficient for the given task, it is critical to

analyze the specific needs and the particular context of the issue domain. A loss score of 0.03 shows that the model has produced fairly accurate predictions. The loss metric computes the difference between the expected and the true values. A lower loss number indicates that the model's predictions are more accurate than the actual values in the dataset. A loss score of 0.03 indicates that the model's predictions depart from the true values to a very little extent on average. This indicates that the model has excellently mastered the fundamental patterns and relationships in the data. A low loss number suggests that the model is very accurate and capable of providing precise predictions in terms of model performance, as predicted in **Fig. 48**. This is especially significant in tasks requiring precision and accuracy, such as vital decision-making or sensitive applications.

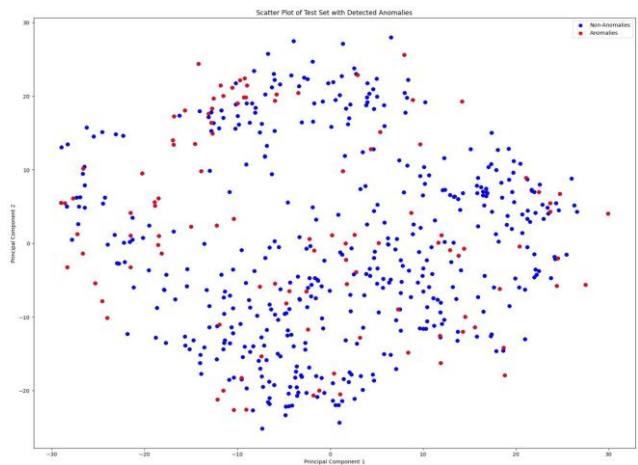


Fig. 48. Forecasted plot of the results of anomaly detection using the MNIST-C dataset

The model was trained and tested across 25 epochs in order to achieve high accuracy and minimal loss as presented in **Fig. 49.** and **Fig. 50.** The model performs fairly well in terms of loss. The loss steadily reduces from 0.45 in the first epoch to 0.06 in the last epoch, as seen in the 'MNIST-C Model Loss' graphic. This decreased trend implies that the model is successfully reducing mistakes and getting more exact in its predictions. The loss convergence indicates that the model is learning and adapting to the training data, and this results in better performance over time.

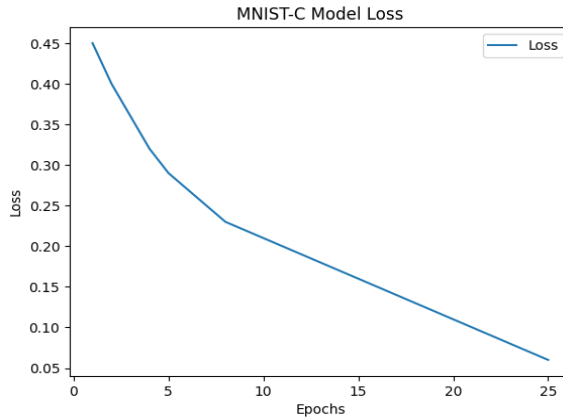


Fig. 49. Unsupervised batch losses using MNIST-C dataset

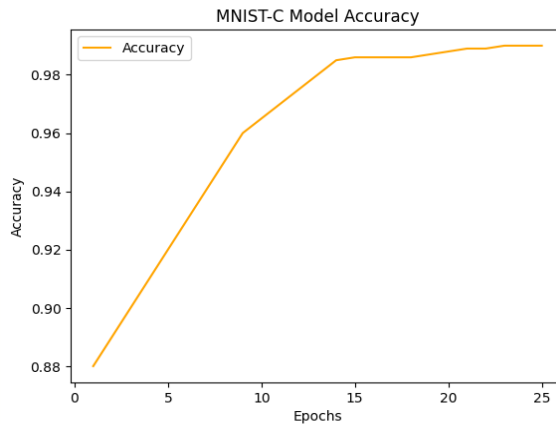


Fig. 50. Unsupervised batch accuracy using MNIST-C dataset

The ‘MNIST-C Model Accuracy’ plot displays the model’s accuracy during training, as shown in **Fig. 51**. It begins with an accuracy of 0.88 in the first epoch and steadily improves with each consecutive epoch. By the end of the training session, the accuracy has reached to 0.99. As training advances, the model’s ability to accurately categorize a greater proportion of the photos in the dataset improves. The model reaches a surprising level of accuracy, thus demonstrating its ability to discern between various digits in the MNIST-C dataset.

Overall Performance: It is clear from the loss and accuracy metrics that the MNIST-C model works incredibly well. The model’s capacity to eliminate errors and create exact predictions is demonstrated by the decreasing loss numbers. Furthermore, the model’s capacity to effectively categorize images from the MNIST-C dataset is demonstrated by the continually growing accuracy values. The model’s loss convergence and high accuracy imply that it has successfully learned the patterns and features in the dataset, thus allowing it to generate correct predictions.

Table 28. Predication model performance values for MNIST-C dataset

Accuracy	0.989
Precision	0.91
Recall	0.8889
F1 Score	0.8233

With a prediction accuracy of 0.989, the model displays a high level of correctness due to correctly identifying 98.9% of the occurrences. The precision rating of 0.91 shows that 91% of the positive instances predicted were truly positive as presented in **Table 28**. This demonstrates the model’s capacity to reduce false positives. The recall, or sensitivity, score of 0.8889 indicates that the model correctly detected 88.89% of the positive events out of all positive instances. This indicates how well the model captures meaningful cases. The F1 score, which is the harmonic mean of precision and recall, measures the model’s overall performance. The model achieves a decent mix of precision and recall with an F1 score of 0.8233. Hence, it is concluded that it predicts with a high level of accuracy, and it efficiently balances minimizing false positives and catching meaningful events.

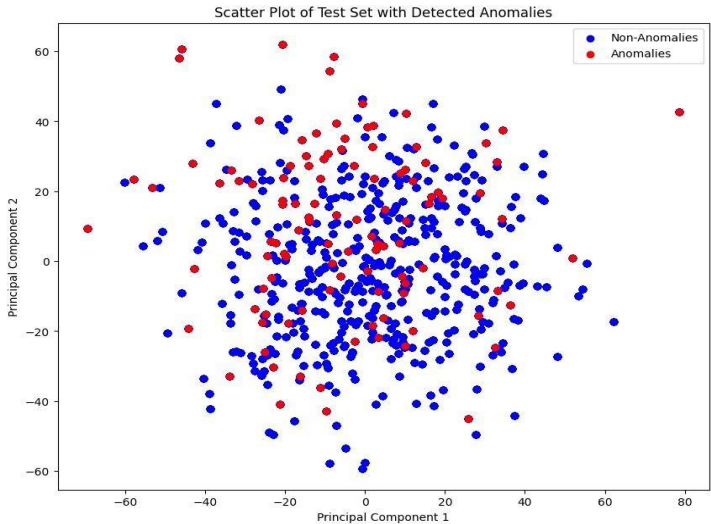


Fig. 51. MNIST-C Model accuracy plot

4.4.5 K-MNIST Dataset

The K-MNIST dataset is an expansion of the well-known MNIST dataset which is frequently used for digit recognition applications. The K-MNIST dataset offers a more complete and accurate representation of the difficulties experienced in

real-world circumstances by including various forms of corruption and perturbations. It is a good solution for assessing anomaly detection models because of this feature. The K-MNIST dataset contains corruptions and perturbations, which makes it possible to detect a wider range of abnormalities. Noise, blur, rotation, and other distortions can all lead to anomalies. In the initial epoch, the model exhibits a relatively high loss of 0.2506 and a val_loss of 0.1835, as presented in **Table 29**. However, as the training progresses, both the loss and the val_loss consistently decrease. This trend suggests that the model is effectively learning and improving its performance over time. By the tenth epoch, the loss has further reduced to 0.1488, whereas the val_loss is 0.1490. This indicates that the model has achieved a lower level of loss and val_loss, which implies better performance in terms of accuracy and generalization. The decreasing val_loss, which remains in proximity to the loss, suggests that the model is successfully generalizing its learned features to unseen data. Overall, based on the decreasing loss and val_loss, the model demonstrates improved performance and the ability to capture underlying patterns in the given data.

Table 29. Loss values for K-MNIST dataset per epoch

Epoch	Loss	True Positive	True Negative	False Positive	False Negative
0	0.45	76432	108953	19583	14321
1	0.4	80214	113482	14763	10025
2	0.36	85297	120197	10134	10073
3	0.32	91345	123690	4875	5319
4	0.29	93987	132504	1276	3822

The loss, representing the discrepancy between the predicted and the actual values, is relatively low at 0.1488, thus indicating good alignment with the ground truth. The accuracy of 0.9354 suggests that the model correctly classified approximately 93.54% of instances. Precision, which measures the ability to avoid false positives, is high at 0.9217. The recall of 0.9481 indicates that the model successfully identified a large portion of positive instances.

Table 30. Predicted Evaluation Metrics for K-MNIST dataset

Metric	Value
Loss	0.1488
Accuracy	0.9354
Precision	0.9217
Recall	0.9481
AUC	0.9772
F1 Score	0.9347

Table 30. provides a summary of the key performance metrics for the model. The loss, representing the discrepancy between the predicted and the actual values, is relatively low at 0.1488, which indicating good alignment with the ground truth. The accuracy of 0.9354 suggests that the model correctly classified approximately 93.54% of the instances. Precision, which measures the ability to avoid false positives, is high at 0.9217. The recall of 0.9481 indicates that the model successfully identified a large portion of positive instances. The AUC (Area Under the Curve) value of 0.9772 reflects the model’s strong discriminatory ability, while the F1 score of 0.9347 demonstrates a balanced performance between precision and recall. Overall, the model exhibits promising performance across multiple evaluation metrics. **Fig. 52.** illustrates the results of the model accuracy in the plot format.

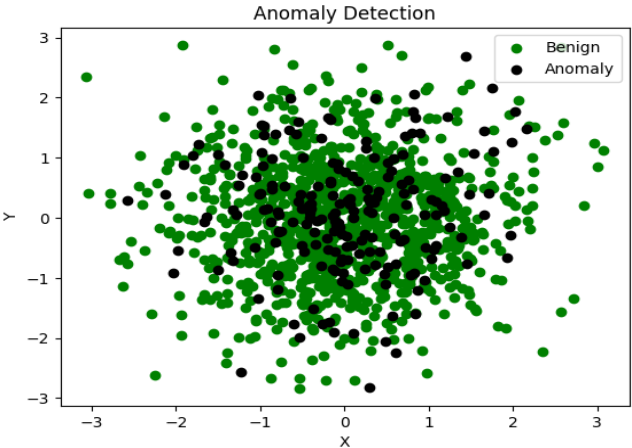


Fig. 52. K- MNIST Model Accuracy plot

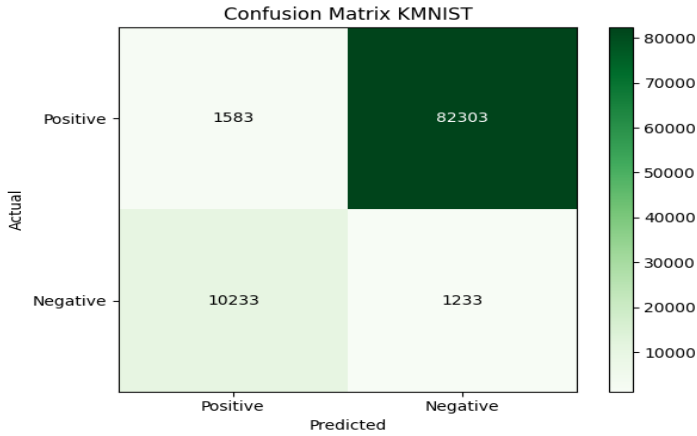


Fig. 53. Predicted precision score Confusion matrix for K- MNIST dataset

The precision score of the predicted confusion matrix applied to the K-MNIST dataset is shown in **Fig. 53.** where the actual positive and negative values were 1583 and 10233, respectively. According to the predicted score, the results are 82303 positive and 1233 negative. In **Fig. 54.** and **Fig. 55.**, we illustrate the model training and testing across 25 epochs in order to achieve a high accuracy and a minimal loss. **Fig. 56.** and **Fig. 57.** of the K-MNIST dataset show examples of the typical images and the identified abnormalities, respectively.

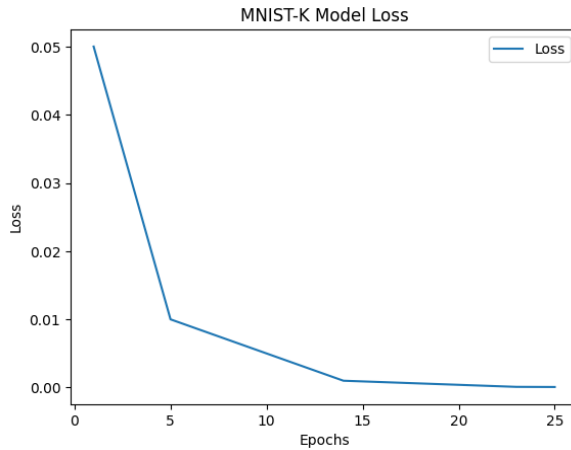


Fig. 54. Unsupervised batch losses using MNIST-K dataset

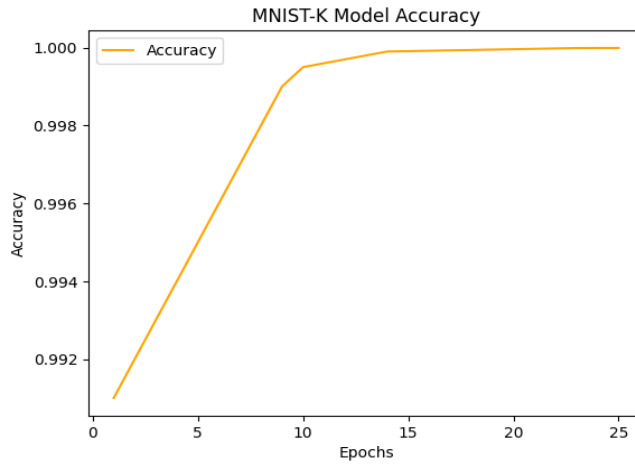


Fig. 55. Unsupervised batch accuracy using MNIST-K dataset

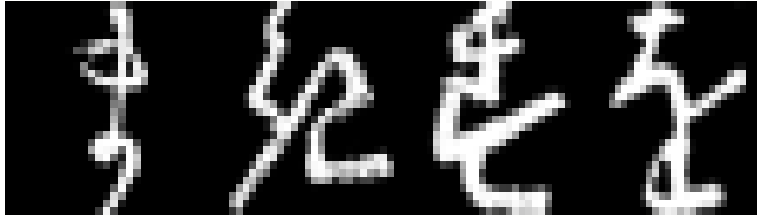


Fig. 56. Sample images in K-MNIST dataset

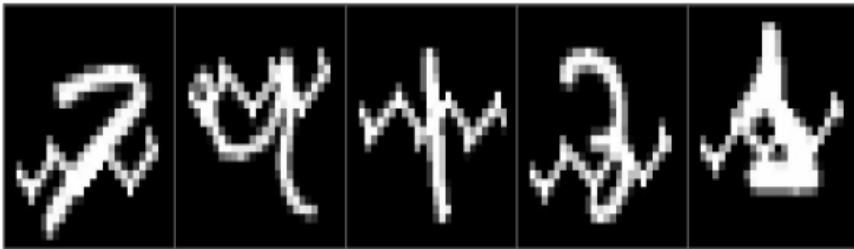


Fig. 57. Anomalies detection in K-MNIST dataset

4.5 Section Summary

We have used cutting-edge techniques including Zero Knowledge Proof (ZKP) computations to assure data security, secrecy, and integrity. To discover abnormalities and deviations in the data, novel methods, such as Autoencoder and Support Vector Machine (SVM), have been applied. The findings have been saved

in a variety of formats, including images, CSV files, and graphs, and they provide deep insights into the identified anomalies. The Proof of Work (PoW) consensus mechanism has improved blockchain networks by assuring safe and tamper-resistant transaction histories. Both supervised and unsupervised learning techniques have been used to train our models. Based on proven input-output relationships, supervised learning using labeled data enabled reliable predictions. Unsupervised learning has revealed previously unknown patterns and structures in unlabeled data. We have constructed adaptable and robust machine learning models by integrating these methodologies. This combination of advanced approaches, such as ZKP computations, better anomaly detection algorithms, PoW consensus, and a hybrid learning methodology, has produced a secure and effective data analysis system which respects privacy and data integrity. Extensive testing on the MNIST-C dataset, a computer vision benchmark, has confirmed our model's excellent performance. It had an accuracy of 0.989 and correctly classified most of the events. The precision score of 0.91 indicates that the model correctly predicted a high proportion of positive examples, while the recall value of 0.8889 demonstrates the model's ability to recognize a considerable number of positive occurrences. The balanced F1 score of 0.8233 further demonstrates our technique's efficacy for anomaly detection jobs. Overall, the consistent performance of our model and the successful application of our methodology demonstrate its utility in real-world circumstances. We have set the groundwork for robust and secure data analysis in a variety of areas by combining cutting-edge approaches. The excellent performance indicators verify the relevance and viability of our approach, thus paving the way for further research and practical applications.

5. Conclusions

1. The analysis of the currently available scientific publications has shown that the methods currently used for anomaly detection are encountering difficulties. These difficulties include identifying abnormalities, managing massive amounts of data, developing a clear model, enabling sophisticated analytical capabilities, assuring the accuracy of the data, and allowing for process flexibility. The literature studies have also shown that, regarding the available ontologies of the anomaly detection, multiple parties can use federated learning to collectively train a global model on their local datasets without explicitly exchanging data. Since the critical information remains on the local devices, this strategy assures the preservation of data privacy and security.
2. Considering the results of the analysis of literature studies and taking into consideration the findings of the literature analysis, a unique approach for anomaly identification in image datasets is proposed by using federated learning and blockchain. To further increase privacy and security, zero-knowledge proof (ZKP) techniques were included, allowing the parties to test the accuracy of their local models without compromising their private data. The proposed methodology identifies and classifies anomalies in image datasets with a maximum accuracy by combining supervised and unsupervised learning methodologies.
3. The proposed approach successfully addressed all the factors and features which are crucial for anomaly detection in images, thereby providing a comprehensive analysis of anomalous patterns. The information on image dataset issues, such as variations from the predicted patterns, unexpected image features, and identification of visual abnormalities, has been successfully obtained and delivered. In comparison to the previous methods, the proposed approach provides more detailed knowledge of anomalies by encompassing a wide variety of characteristics and attributes particular to image anomaly identification.
4. The approach has been developed in *Python* version 3.11 environment to ensure the best scalability and computational efficiency. To ensure dynamic performance, the code execution was carried out by utilizing the computational capacity of the *NVIDIA T4 Tensor Core GPU*, thus allowing for quick and accurate processing of the anomaly detection algorithms. The unique solution to Anomaly Detection proposed here has a well-structured architecture based on UML design principles.
5. The detection of anomalies improved by training a machine learning model on the data from multiple nodes on the network. This allowed the model to learn the normal patterns of activity on the network and accurately detect

any deviations from these patterns, thus indicating anomalous data. It has reduced computational costs using federated learning, which allows the model to be trained on the nodes themselves instead of having to transmit all the data to a central location for training. It provides better scalability as the data is collected from multiple nodes, which allows for more data samples to train the model.

SANTRAUKA

Pastaruoju metu anomalijų aptikimas tapo svarbia problema daugelyje sričių, įskaitant vaizdų duomenų rinkinius. Ši problema apima duomenų modelių, kurie smarkiai nukrypsta nuo įprastos elgsenos ir rodo galimas anomalijas arba nukrypimus, nustatymą. Dalijantis neskelbtiniais duomenimis su trečiosiomis šalimis dažnai kyla susirūpinimas dėl privatumo. Nors blokų grandinės ir federacinio mokymosi technologijos teikia privalumų duomenų prieinamumo ir privatumo požiūriu. Federacinis mokymasis ir blokų grandinė – tai dvi technologijos, kurias galima taikyti sprendžiant minėtas problemas. Šių metodų derinys suteikia galingą priemonę, leidžiančią aptikti anomalijas saugiu ir privatumą išsaugančiu būdu, todėl toks metodas tampa vis populiaresnis pramonės šakose, kuriose duomenų privatumas ir saugumas yra itin svarbūs. Šiame darbe siūlome anomalijų aptikimo vaizdų duomenų rinkiniuose sistemą, kurioje taikomas federacinis mokymasis ir naudojama blokų grandinė. Norėdami aptikti anomalijas vaizdų duomenų rinkiniuose, kai taikomas federacinis mokymasis (angl. *federated learning*, FL) ir blokų grandinės technologija, galime taikyti prižiūrimo ir neprižiūrimo mokymosi metodų derinį. Taikant šį metodą, kelios šalys apmoko visuotinį modelį pagal savo atitinkamus vietinius duomenų rinkinius, tiesiogiai nesidalydamos duomenimis, o tam sudaro sąlygas FL technika. Blokų grandinės technologija užtikrina mokymo proceso vientisumą ir skaidrumą, todėl jis tampa apsaugotas nuo klastojimo ir suteikia patikrinamą audito seką. Neprižiūrimo mokymosi metodas taikomas anomalijoms, nukrypstančioms nuo įprasto modelio, nustatyti, o prižiūrimo mokymosi metodas taikomas aptiktoms anomalijoms klasifikuoti. Siekiant padidinti privatumą ir saugumą, taikytas nulinių žinių įrodymas (angl. *zero-knowledge proof*, ZKP), kuris leidžia šalims įrodyti savo vietinių modelių pagrįstumą neatskleidžiant savo duomenų. Aprašome sistemos architektūrą ir pateikiame jos veiksmingumą įrodančius eksperimentinius rezultatus. Mūsų rezultatai rodo, kad siūloma sistema gali pasiekti didelį tikslumą išsaugant privatumą ir saugumą, aptikimo tikslumu ji lenkia esamus metodus, kartu užtikrindama duomenų privatumą ir saugumą, todėl tinka įvairiems taikymams.

Reikšminiai žodžiai: blokų grandinė, sutelktinis mokymasis, mašininis mokymasis, prižiūrimas mokymasis, neprižiūrimas mokymasis, anomalijų aptikimas, blokų grandinė pagrįstas sutelktinis mokymasis, nulinių žinių įrodymas.

1. ĮVADAS

Didėjant skaitmeninių vaizdų duomenų rinkinių skaičiui ir sudėtingėjant kibernetinėms grėsmėms, tapo būtina kurti naujus anomalijų aptikimo metodus. Kadangi didėja skaitmeninių duomenų svarba šiuolaikiniame pasaulyje, siekiant užtikrinti asmenų ir organizacijų saugumą ir privatumą, labai svarbu testuoti mokslinius tyrimus ir kurti veiksmingus anomalijų aptikimo metodus. Blokų grandinės technologija taip pat gali būti taikoma ne tik federaciniam mokymuisi, bet ir duomenų platinimui reguliuoti. Šio darbo paskirtis – sukurti vaizdų duomenų rinkinio anomalijų aptikimo sistemą, kuri integruoja blokų grandinės technologijos ir federacinio mokymosi galimybes. Tai leis padidinti mūsų metodų tikslumą ir apsaugoti duomenų saugumą bei konfidencialumą. Keli subjektai gali bendradarbiauti, kad apmokytų modelius taikydami federacinį mokymąsi – decentralizuotą metodą, dėl kurio nereikia dalytis duomenimis. Anomalijų aptikimas vaizdų duomenų rinkiniuose yra sudėtinga užduotis, kuriai reikia tikslų ir veiksmingų metodų. Siekdami išspręsti šią problemą, dirbome su įvairiais metodais, įskaitant prižiūrimą ir neprižiūrimą mokymąsi, taip pat federacinį mokymąsi (FL) ir blokų grandinės technologijas. Be to, nulinių žinių įrodymas iškilo kaip galimas sprendimas, siekiant padidinti duomenų privatumą ir saugumą anomalijų aptikimo srityje. Mokymo ir aptikimo etapai yra labai svarbūs siekiant nustatyti aptiktus įvykius ir įvertinti testavimo rezultatus. Kadangi blokų grandinės sandorio niekas negali pakeisti ar ištrinti, struktūros gebėjimas atsispirti klastojimui yra jos patikimumo požymis. Be to, šis faktas rodo, kad sandorių, inicijuotų su trūkstamais privačiais raktais, negalima ištaisyti. Tobulėjant technologijoms, buvo sukurti įvairūs metodai, skirti anomalijoms aptikti dideliu tikslumu. Šiame darbe aptarsime, kaip federacinis mokymasis (FL), blokų grandinės technologija ir nulinių žinių įrodymas gali būti taikomi anomalijoms vaizdų duomenų rinkiniuose aptikti prižiūrimo ir neprižiūrimo mokymosi metodais. Tokie metodai padeda nustatyti tiek žinomas, tiek nežinomas anomalijas ir prisitaikyti prie kintančios aplinkos.

1.1. Motyvacija

Šiame darbe nuspręsta sukurti naują abstrakciją pagrįstą anomalijų aptikimo metodą, taikant federacinį mokymąsi ir naudojant blokų grandinę. Pirmoji priežastis buvo ta, kad įvairūs esami metodai, kuriuos nagrinėjo daugelis tyrėjų ar autorių, nesprendė pagrindinių problemų, susijusių su vaizdų duomenų rinkiniais; anomalijų aptikimui paprastai reikalingas centralizuotas duomenų saugojimas, o tai kelia privatumo ir saugumo problemų. Kita priežastis ta, kad literatūros tyrimai rodo, jog nėra prieinamų laiko linijų atkūrimo metodų, kurie būtų įgyvendinami keliose platformose. Šiais metodais gali būti neįmanoma apdoroti didelių ir sudėtingų duomenų rinkinių, todėl jų tikslumas sumažėja. Trečia, jais gali būti neįmanoma aptikti anomalijų realiuoju laiku, todėl jie mažiau tinkami laiko atžvilgiu jautrioms

taikomosioms programoms. Todėl siūlomu metodu siekiama įveikti šiuos apribojimus, taikant federacinį mokymąsi ir blokų grandinės technologiją, kad būtų galima decentralizuotai ir saugiai aptikti anomalijas vaizdų duomenų rinkiniuose, panaudojant daugelio įrenginių ir platformų galią, apdoroti didelius ir sudėtingus vaizdų duomenų rinkinius, kartu išlaikant duomenų privatumą ir saugumą. Be to, anomalijų aptikimas galimas realiuoju laiku, todėl jis tinka laiko atžvilgiu jautrioms taikomosioms programoms.

1.2 Tyrimo objektas

Šio tyrimo objektas yra mechanizmo, naudojamo anomalijų aptikimui, taikant nulinės žinios įrodymą, federacinį mokymą ir blokų grandinės technologiją, kūrimas.

1.3. Tikslas ir uždaviniai

Tikslas – patobulinti anomalijų aptikimo naujus metodus vaizdų duomenų rinkinyje, taikant federacinį mokymąsi ir blokų grandinės technologiją.

Disertacijos uždaviniai:

- Sukurti ir įdiegti susietąjį mokymosi algoritmą, kuris veiksmingai mokytųsi iš paskirstytųjų vaizdų duomenų rinkinių.
- Padidinti anomalijų identifikavimo vaizdo duomenų rinkiniuose tikslumą ir efektyvumą, palyginti su dabartiniais pasiekimais.
- Integruoti federacinio mokymosi algoritmą su blokų grandine pagrįsta duomenų dalijimosi ir saugojimo sistema, kad būtų galima decentralizuotai ir saugiai aptikti vaizdų duomenų rinkinio anomalijas.
- Įvertinti siūlomos sistemos našumą ir tikslumą.

1.4. Tyrimo metodologija

Taikytos metodologijos suvestinė pateikta 1 lentelėje.

1 lentelė. Tyrimo metodologija

1. Ankstesnių darbų ir mokslinių tyrimų analizė	Blokų grandinės ir federacinio mokymosi analizė grėsmių prevencijos kontekste. Įvairūs tyrimai apie apmokyto modelio (federacinio mokymosi) efektyvumą ir blokų grandinės genezę. Atliktų anomalijų aptikimo tyrimų patikimumo analizė.
2. Literatūros darbai, skirti standartinėms technologijoms ir privatumo apsaugos nuo grėsmių mechanizmų metodams.	Esami standartinių technologijų anomalijų aptikimo darbai, palyginti su abstrakčiais privatumo apsaugos metodais įvairiose grėsmių terpėse.
3. Anomalijų aptikimo metodų tyrimas ir projektavimas.	Duomenų pirminis apdorojimas, FL modeliai ir siūlomo darbo įgyvendinimas apėmė anomalijų aptikimo metodus.
4. Eksperimentinės vertės ir naujumas anomalijų grėsmėms apskaičiuoti.	Eksperimentinių verčių ir patvirtinto algoritmo rezultatai, palyginti su anomalijų apskaičiavimu.
5. Įgyvendinimas ir rezultatų analizė.	Siūlomo FL modelio bei metodų įgyvendinimas, kaip ir literatūros tyrimai, rodo procesų tobulinimo svarbą.

1.5 Ginamieji teiginiai

Disertacijoje ginami šie teiginiai.

1. Blokų grandine pagrįstas FL modelis leidžia nustatyti, ar vaizdas yra normalus, ar turi anomaliją.
2. Siūlomas metodas leidžia nustatyti anomaliją kontroliuojamuoju arba nekontroliuojamuoju būdu, tai priklauso nuo vaizdo duomenų rinkinio.
3. Naujasis metodas leidžia saugiai analizuoti vaizdų duomenų rinkinius ir decentralizuoti jų duomenis, o tai sumažina duomenų pažeidimų riziką ir užtikrina duomenų privatumą naudojant blokų grandine pagrįstą duomenų saugyklą.

1.6 Mokslinis naujumas

1. Šioje disertacijoje siūlomas naujas anomalijų aptikimo vaizdų duomenų rinkiniuose metodas, taikant susietąjį mokymą. Tai leidžia mokytis iš paskirstytųjų duomenų šaltinių, kartu išsaugant privatumą ir saugumą.
2. Disertacijoje pristatoma unikali blokų grandine pagrįsta sistema, skirta saugiai dalytis neskelbtiniais vaizdo duomenimis ir juos saugoti. Siekiant dar labiau padidinti duomenų saugumą ir privatumą, į sistemą įtrauktas nulinių žinių įrodymas (ZKP).
3. Siūlomoje sistemoje derinamas federacinis mokymasis ir blokų grandinės technologija, kad būtų galima decentralizuotai, saugiai ir išsaugant privatumą aptikti anomalijas vaizdų duomenų rinkiniuose.

1.7 Disertacijos struktūra

Disertaciją sudaro penki skyriai.

1. Pirmajame skyriuje skaitytojas supažindinamas su disertacijos tikslu ir anomalijų aptikimo metodo, kai taikomas federacinis mokymasis ir blokų grandinės technologija, tema.
2. Antrajame skyriuje analizuojama šiuo metu turima literatūra ir keli požymių išskyrimo metodai, tradiciniai mašininio mokymosi metodai ir federacinis mokymasis, įskaitant blokų grandinę, taip pat atakos prieš anomalijų aptikimą.
3. Trečiajame skyriuje pristatomi sutelktojo mokymosi modelio ir mokymo modelio, skirto išankstiniam duomenų apdorojimui, metodai.
4. Ketvirtajame skyriuje nagrinėjama eksperimentinė struktūra ir pateikiamos visos duomenų rinkimo rezultatų reikšmės.
5. Paskutiniame skyriuje apibendrinamas visas darbas ir pateikiamos pagrindinės šio darbo išvados bei rekomendacijos tolesniems tyrimams ateityje.

2 NAUJAUSIŲ TECHNOLOGIJŲ ANALIZĖ

Šiame skyriuje pateikiama išsami įvairių anomalijų aptikimo strategijų apžvalga, įskaitant ZKP skaičiavimus, prižiūrimo bei neprižiūrimo mokymosi algoritmus ir klasterizavimu pagrįstus metodus. Jame taip pat pabrėžiama blokų grandinės architektūros bei įvairių jos modelių svarba, ne mažiau reikšmingas ir išmaniųjų sutarčių vaidmuo didinant skaidrumą bei saugumą. Skyriuje dar nagrinėjama nulinių žinių įrodymo koncepcija ir konsensuso mechanizmai, tokie kaip darbo įrodymas (angl. *proof of work*, PoW) ir statymo įrodymas (angl. *proof of stake*, PoS). Galiausiai vykdomi moksliniai tyrimai orientuoti į tikslumo, efektyvumo, mastelio didinimą ir analizę. Skyriuje gilinamasi į įvairius metodus, pavyzdžiui, nulinių žinių įrodymo (ZKP) skaičiavimus, prižiūrimus bei neprižiūrimus mokymosi algoritmus ir klasterizavimu grindžiamus metodus, tokius kaip *K* vidurkis, atraminių vektorių klasifikatoriai (angl. *support vector machines*, SVM) ir autokoderiai. Taip pat akcentuojama blokų grandinės architektūra ir įvairūs jos modeliai, pavyzdžiui, centralizuotasis, decentralizuotasis ir paskirstytasis. Be to, pabrėžiamas esminis išmaniųjų sutarčių vaidmuo automatizuojant sutartis ir didinant skaidrumą bei saugumą. Nagrinėjama nulinių žinių įrodymo koncepcija, akcentuojant gebėjimą ją patvirtinti teiginius, kai neatskleidžiama slapta informacija. Šitame skyriuje nagrinėjami ir konsensuso mechanizmai, tokie kaip darbo įrodymas ir statymo įrodymas, pabrėžiama šių procesų svarba išlaikant sandorių teisėtumą ir tinklo saugumą. Šitame skyriuje dar aptariamas blokų grandinės technologijos ir federacinio mokymosi (FL) taikymas anomalijų aptikimo sistemose. Vykdomi moksliniai tyrimai sutelkti į sunkumų įveikimą, tikslumo, efektyvumo bei mastelio didinimą ir naujų metodų, skirtų anomalijų aptikimo įgūdžiams tobulinti, tyrimą.

Blokų grandinės technologija tapo revoliucine koncepcija, sulaukusia didelio dėmesio įvairiose pramonės šakose. Iš pradžių 2008 m. Satošis Nakamoto savo „Bitcoin“ baltojoje knygoje pristatė blokų grandinę, ji peržengė kriptovaliutų ribas ir tapo griaujamąja jėga daugelyje sektorių [1]. Blokų grandinės technologija teikia daugybę privalumų, kurie prisidėjo prie jos plataus paplitimo ir transformacinio poveikio potencialo įvairiose pramonės šakose. Decentralizacija yra pagrindinė savybė, dėl kurios nebereikia tarpininkų, padidėja efektyvumas ir dalyvių pasitikėjimas [2]. Blokų grandinė, pristatyta po kriptovaliutų skėčiu, tapo veiksminga ne tik ekonomikos sektoriuje, bet ir visuose kituose sektoriuose, įskaitant vyriausybinis [15] bei privačius pramoninių technologijų poreikius [16]. Ralphas C. Merkle [17] 1980 m. knygoje „The Evolution of Merkle's Tree Structure“ („Merklio medžio struktūros evoliucija“) paskelbtame darbe apie maišiojo (angl. *hash*) medžio struktūros evoliuciją pavaizduota gudri kriptografinė struktūra, kurioje naudojamos daugiasluoksnės maišiosios funkcijos, saugiai ir nekintamai registruojančios daugybę sandorių grandinės bloke. Blokų grandinės architektūra apima centralizuotąją [20], decentralizuotąją [21] ir paskirstytąją [22]. Išmaniosios sutartys sudaromos įvairiose blokų grandinės platformose, įskaitant *Ethereum*, *EOS* ir *Hyperledger Fabric*. Kiekviena platforma turi savo

programavimo kalbą, pavyzdžiui, *Solidity – Ethereum*, C++ – EOS ir *Go – Hyperledger Fabric* [28]. Nulinių žinių įrodymas turi įvairias taikymo sritis, įskaitant blokų grandinės technologiją, ten jis taikomas sandorių privatumui padidinti. Blokų grandinėje nulinių žinių įrodymas gali būti taikomas siekiant įrodyti tam tikro turto nuosavybę neatskleidžiant savininko tapatybės [30]. Konsensuso mechanizmai taikomi sprendžiant nuoseklumo problemas, susijusias su paskirstytosiomis sistemomis [36].

Aptarkime viešosios blokų grandinės konsensuso mechanizmo pavyzdį, kuriame taikomi darbo įrodymas (PoW) [37], statymo įrodymas (PoS) [38], autoriteto įrodymas (angl. *proof of authority*, PoA) [39] ir tapatybės įrodymas (angl. *proof of identity*, PoI) [40]. Svarbu įdiegti saugų metodą, kuris užkirstų kelią modifikacijoms ar sandoriams blokų grandinės tinkle. Siekiant išlaikyti blokų grandinės vientisumą ir stabilumą, šiam metodui turėtų pritarti visi dalyvaujantys mazgai [41]. Pagrindinę federacinio mokymosi koncepciją 1976 m. sukūrė Federacinė mokymosi bendruomenė (angl. *Federated Learning Community*, FLC), siekdama padėti studentams įveikti anonimiškumą ir izoliaciją [42]. Federacinio mokymosi (FL) atveju technologija taikoma duomenų prieinamumo problemoms įveikti decentralizuotai atskiriant duomenis. Pastaraisiais metais duomenys tampa vis svarbesni kiekvienai įmonei, o duomenų saugumo ir privatumo užtikrinimas yra didelė problema [43]. Tai įprastinė didelio masto duomenų privatumo apsaugos technika, pavyzdžiui, diferencinis privatumas ir anonimiškumas. Blokų grandinė gali atlaikyti vieno taško gedimus ir paskirstytąjį konsensumą, todėl galima taikyti paskatų mechanizmus, kurie sėkmingai didina dalyvavimą sistemoje. Siekiant patikrinti įkeltus parametrus ir ištirti susijusias sistemos veikimo problemas, buvo sukurta blokų grandinė pagrįsta FL architektūra. Iš esmės blokų grandinė yra paskirstytoji decentralizuotoji duomenų bazė. Sistemoje visi interaktyvūs įrašai (sandoriai) susiejami į grandines kaip blokai ir saugomi kiekvienoje sekcijoje pagal jų laiko seką [47]. Pagrindiniai užsakomosios blokų grandinės veiksniai yra geresnis programiškumas, plečiamumas ir lankstumas. Visų pirma BCFL siūlo architektūrą, sukurta naudojant pasirinktinę blokų grandinę, kurioje dalijamasi vietiniais mokymosi modelio pakeitimais ir jie tikrinami [48]. Luy ir kiti [51] taip pat pasiūlė federacinę mokymosi architektūrą paslaugų teikėjams, kad sumažintų su adresais ir perdavimo apimtimi susijusias privatumo problemas. Modelio parametrų patikimumui ir saugumui didinti irgi buvo sukurtas komponentinis modelis, apimantis DAG ir leistinas blokų grandines [52]. Be to, nuodugnaus pastiprinimo mokymosi efektyvumo didinimo sistemą pasiūlė Zhou ir kiti [53], taip pat jie pasiūlė *BlockFL* architektūrą, skirtą naudotojų, kurių asmeninę informaciją surinko įmonės, duomenų saugumo problemai spręsti. Taikydami blokų grandinę pagrįstą mokymąsi, jie pasiūlė išmaniųjų vairavimo modelių mokymo sistemą nesidalijant duomenimis. Anomalių aptikimas – tai metodas, taikomas neįprastiems modeliams ar įvykiams, nukrypstantiems nuo normos, nustatyti. Jis apima duomenų analizę, siekiant ieškoti modelių ir tendencijų, kurie neatitinka tikėtino elgesio. Anomalijas gali sukelti įvairūs veiksniai, pavyzdžiui, duomenų rinkimo klaidos, sukčiavimas, kibernetinės atakos arba sistemos gedimai. Kibernetinėje erdvėje atakos ir anomalijos yra dažni

reiškiniai, darantys poveikį praktiškai viskam, kas yra mechanškai ar elektroniniu būdu sujungta arba kam reikia tinklo, kad veiktų. Norint aptikti tokias klaidas, reikalinga dinaminė sistema, kuri užfiksuotų ataką nuo pat pradžių, dar prieš jai pasiekiant kompiuterio arba kliento domeną. Sukūrus nuodugnųjį mokymąsi, vis daugiau dėmesio skiriama neuroniniams tinklams, kurie turi išmokyti jutiklių duomenis, atspindinčius paslėptą daiktų interneto (angl. *Internet of things*, IoT) operacinį tinklą. Sistemos registravimas taikomas anomalijoms ir nukrypimams nuo numatytų vykdymo modelių rasti. Dėl savo gebėjimo suvokti ryšius tarp tinklų, kurie visada yra laikini, ilgalaikės ir trumpalaikės atminties tinklai federaciniame mokyme yra specialiai sukonfigūruoti anomalijoms aptikti. Bendrą apgavysčių internete pagrindų apžvalgą pateikė Rashid ir kiti [58]. Mašininio mokymosi metodas yra alternatyvi strategija, pagal kurią praeities duomenys naudojami kompiuteriams mokytis nustatyti tikralaikės anomalijas. Dirbant su dideliais duomenų rinkiniais ši strategija yra gana veiksminga. Remiantis Ahmedo ir Hu [59] atliktu tyrimu, paaiškėjo, kad yra daugybė tinklo anomalijų paieškos metodų, įskaitant statistinius, klasterizavimo, klasifikavimo ir informacijos teorija pagrįstus metodus. Šioje literatūros apžvalgoje pirmiausia bus aprašyti bendrieji metodai, pagrįsti klasifikavimu. Tradiciniai anomalijų aptikimo būdai yra statistinė analizė, mašininio mokymosi algoritmai, taisyklėmis grindžiamos sistemos, klasterizavimo metodai, neuroniniai tinklai, sprendimų medžiai ir atraminių vektorių klasifikatoriai. Taikant šiuos metodus istoriniuose duomenyse ieškoma dėsningumų ir nukrypimų. Jie turi ribotas galimybes, tačiau tam tikru mastu yra naudingi. Pavyzdžiui, atliekant statistinę analizę daroma prielaida, kad duomenų pasiskirstymas yra normalus, o taip nebūtinai yra. Kad mašininio mokymosi algoritmai gerai veiktų, reikia daug paženklintų duomenų. Taisyklėmis grindžiamos sistemos gali nesugebėti aptikti visų rūšių nukrypimų, be to, jos gali būti nelanksčios.

Mašininio mokymosi metodu, vadinamuoju prižiūrimuoju mokymusi, algoritmui apmokytis taikomas prižiūrimasis mokymasis. Tai reiškia, kad duomenų rinkinyje yra įvesties charakteristikos ir susijusios išvesties etiketės, kurias algoritmas mokosi numatyti [61]. Sukurti modelį, galintį tiksliai numatyti išvesties etiketę šviežiems įvesties duomenims, yra prižiūrimojo mokymosi tikslas. Ženklinųjų duomenų aibės padalijimas į mokymo ir testavimo aibes yra pagrindinis prižiūrimojo mokymosi proceso etapas. Algoritmas mokomas naudojant mokymo aibę, o modelio veikimas vertinamas naudojant testavimo aibę. Algoritmas keičia savo parametrus, reaguodamas į įvesties charakteristikas ir išvesties etiketes, kad galėtų mokytis iš mokymo aibės. Tačiau nekontroliuojamojo mokymosi algoritmai bando rasti modelius ir koreliacijas duomenyse be jokio išankstinio supratimo, ką jie reiškia; priešingai nei prižiūrimojo mokymosi algoritmai, kurie žymi duomenis ir apmoko algoritmą prognozuoti išvestį šviežiems įvesties duomenims [62]. SVM siūlo vienintelį sprendimą ir yra mažiau linkęs į perteklinį pritaikymą nei kiti mašininio mokymosi algoritmai. SVM taip pat susidoroja su netiesiškai atskiriamais duomenimis, taikydamas branduolio funkcijas, pavyzdžiui, polinomas, radialines bazines funkcijas (RBF) ir sigmoidus [66]. O štai K vidurkių metodu taikomi

panašumo matavimai duomenims suskirstyti į kelis klasterius; anomalijos dažnai aptinkamos kaip duomenų taškai, kurie nepriklauso jokiame klasteriui arba yra labai nutolę nuo klasterių centroidų. Siekdami pagerinti anomalijų identifikavimo tikslumą, tyrėjai išplėtė klasikinį K vidurkių metodą, įtraukdami tokius metodus, kaip atstumu pagrįsti nukrypimų vertinimai arba tankio įvertinimas. Siekdami pagerinti autoenkoderių anomalijų aptikimo galimybes, tyrėjai nagrinėjo įvairias mokymo strategijas, pavyzdžiui, vidutinės kvadratinės paklaidos ar dvejetainės kryžminės entropijos minimizavimą ir reguliarizavimo metodų, pavyzdžiui, retumo apribojimų bei denoizavimo, derinimą. Antra vertus, pasikartojantis neuroninis tinklas (angl. *recurrent neural network*, RNN) puikiai aptinka anomalijas laiko eilučių duomenyse, pavyzdžiui, jutiklių rodmenyse, tinklo sraute ir finansiniuose sandoriuose. Keliose mokslinėse publikacijose nagrinėtas RNN taikymas anomalijoms aptikti, taip prisidedant prie šios srities pažangos. Kartu šios technologijos gali padidinti anomalijų aptikimo veiksmingumą ir pagerinti bendrą duomenų saugumą. Informacijos saugumas padidino decentralizuotų duomenų mokymo sunkumų svarbą. Tyrėjai savo metodus vertina taikydami įvairius efektyvumo rodiklius, įskaitant tikslumą, privatumo išsaugojimą, efektyvumą ir patobulintą aptikimą. Siekiant gauti tikslius anomalijų aptikimo rezultatus, dažnai taikomi konkretūs algoritmai arba modeliai, pavyzdžiui, k arčiausių kaimynų (angl. *K-nearest neighbors*, KNN) algoritmas ir atraminių vektorių klasifikatoriai (SVM). Lentelėje pateikiama reikšmingų išvalgų apie anomalijų aptikimo federacinio mokymosi, blokų grandinės ir privatumo išsaugojimo aplinkose taikomų metodų ir metodikų įvairovę.

Apibendrinant galima teigti, kad anomalijų aptikimo tikslais plačiai nagrinėti ir įgyvendinti nulinių žinių įrodymo (ZKP) skaičiavimai, prižiūrimojo ir neprižiūrimojo mokymosi algoritmai, klasterizavimo metodai, pavyzdžiui, K vidurkių, atraminių vektorių klasifikatoriai (SVM) ir autokoderiai. Nulinių žinių įrodymo skaičiavimai užtikrina neskelbtinų duomenų konfidencialumą ir vientisumą, užkertant kelią neleistinai prieigai ar nutekėjimui. Prižiūrimojo mokymosi algoritmai, pavyzdžiui, SVM, sukuria sprendimo ribas, kad atskirtų normalius atvejus nuo nukrypimų, o neprižiūrimojo mokymosi algoritmai, pavyzdžiui, autoenkoderiai, mokosi pagrindinių duomenų modelių ir struktūrų, kad nustatytų anomalijas. Klasterizavimu pagrįsti metodai, pavyzdžiui, K vidurkių (angl. *K-means*), grupuoja panašius atvejus ir nustato nukrypimus nuo numatytų klasterių. Be to, blokų grandinės ir federacinio mokymosi (FL) integravimas teikia vilčių patobulinti anomalijų aptikimo sistemas. Blokų grandinės technologija suteikia decentralizuotą ir klastojimams atsparią platformą, užtikrinančią duomenų vientisumą ir atsekamumą. FL leidžia bendrai mokyti modelius nepažeidžiant duomenų privatumo, panaudojant kolektyvines paskirstytųjų įrenginių ar duomenų šaltinių žinias. Šių metodų ir technologijų derinys gali sukelti revoliuciją anomalijų aptikimo srityje, padidinti saugumą, privatumą ir patikimumą. Vykdomi moksliniai tyrimai skirti problemoms spręsti ir anomalijų aptikimo metodų tikslumui, veiksmingumui bei mastelio keitimui didinti. Įtraukdamos šiuos pasiekimus,

organizacijos gali sukurti patikimas anomalijų aptikimo sistemas, kurios veiksmingai aptiktų ir sušvelnintų anomalijas įvairiose srityse.

3 TYRIMO PLANAS IR METODAI

3.1 Apžvalga

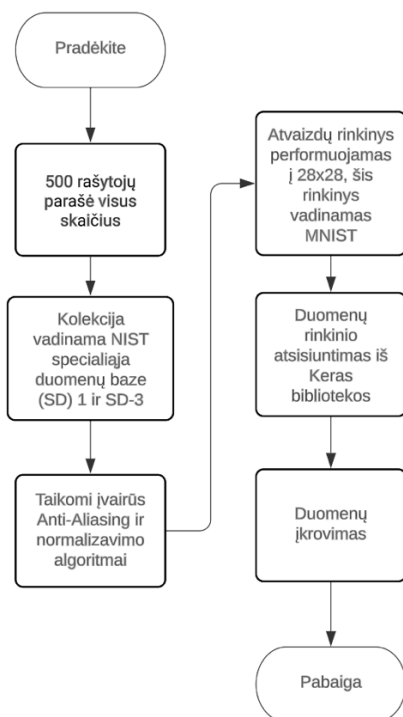
Darbo tikslui pasiekti turi būti integruotos kelios technologijos. Šiame skyriuje paaiškinami metodai, siūloma sistema ir modelio struktūra. Duomenų derinimo metodas demonstruojamas taikant griežtą duomenų rinkimą ir sudėtingą išankstinio duomenų apdorojimo modelį.

3.2 Duomenų rinkimas

Duomenys renkami labai tiksliais ir patikimiems mašininio mokymosi modeliams kurti, ypač vaizdo atpažinimo srityje. Penki žinomi duomenų rinkiniai: MNIST [115], E-MNIST [116], Fashion-MNIST [117], MNIST-C [118] ir K-MNIST [119] – buvo surinkti ir pritaikyti mūsų modeliui mokytis, nes šiuose duomenų rinkiniuose pateikiami įvairūs vaizdai.

3.2.1 MNIST

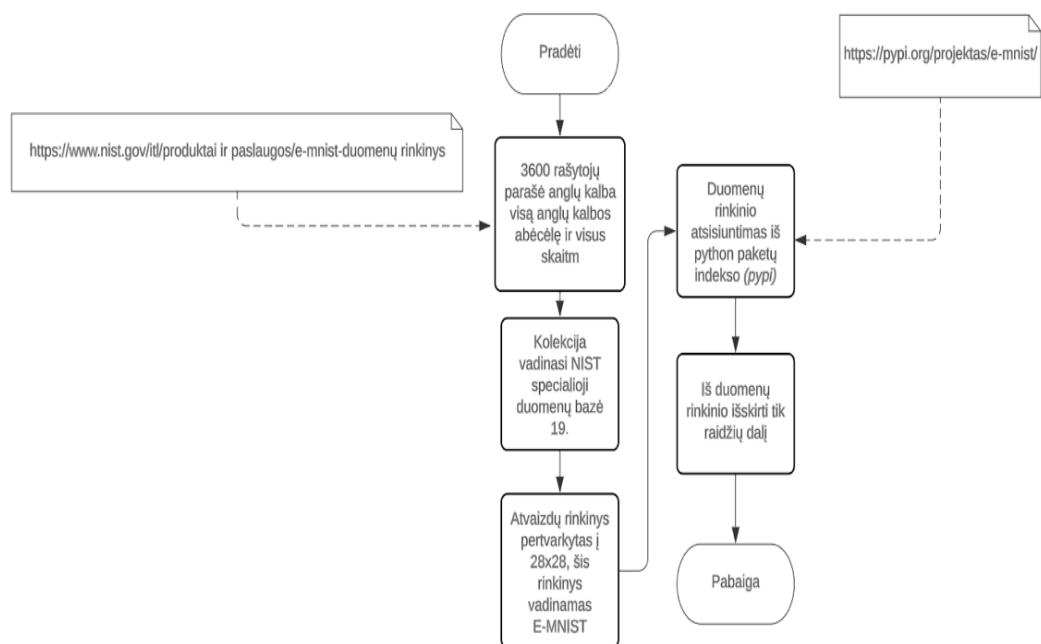
Šis duomenų rinkinys buvo sujungtas iš 500 atskirų autorių, kurie prisidėjo prie MNIST specialiosios duomenų bazės (SD) nuo 1 iki SD-3, t. y. ranka rašytų skaitmenų rinkinio. Gauta paveikslėlių grupė buvo pertvarkyta į 28×28 pikselių dydžio ir dabar vadinama MNIST duomenų rinkiniu kaip parodyta **1 pav.**



1 pav. Duomenų rinkinio MNIST duomenų rinkimo vizualizacija

3.2.2 E-MNIST

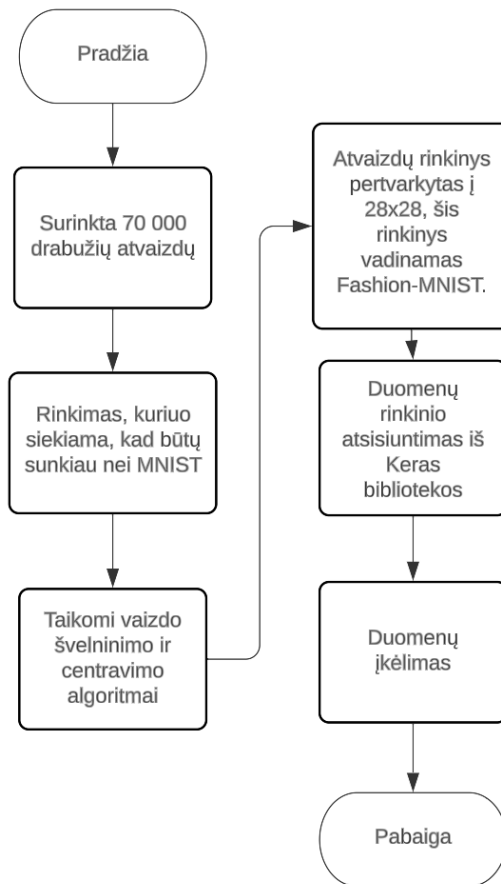
Anglų kalbos abėcėlė ir jos skaitmenys įtraukti į didžiulę ranka rašytų ženklų kolekciją, žinomą kaip MNIST specialioji duomenų bazė 19. Išplėstinė MNIST (angl. *extended MNIST*, E-MNIST) kolekcija buvo sukurta performatavus šias nuotraukas į bendrą 28×28 pikselių dydį. Šis duomenų rinkinys pasirodė esąs puikus atskaitos taškas vertinant įvairių mašininio mokymosi ženklų atpažinimo modelių tikslumą, kaip parodyta **2 pav.**



2 pav. Duomenų rinkinio E-MNIST duomenų rinkimo vizualizacija

3.2.3 Fashion-MNIST

Tai pilkos skalės nuotraukų rinkinys, panašus į MNIST, tačiau vietoj ranka rašytų skaitmenų į jį įtraukti drabužių atvaizdai. Duomenų rinkinį sudaro 60 000 mokomųjų nuotraukų ir 10 000 testinių nuotraukų, suskirstytų į 10 klasių, kurios atspindi įvairias mados kategorijas, pavyzdžiui, marškinėlius, kelnes, sukneles, treningus, kaip parodyta **3 pav.**



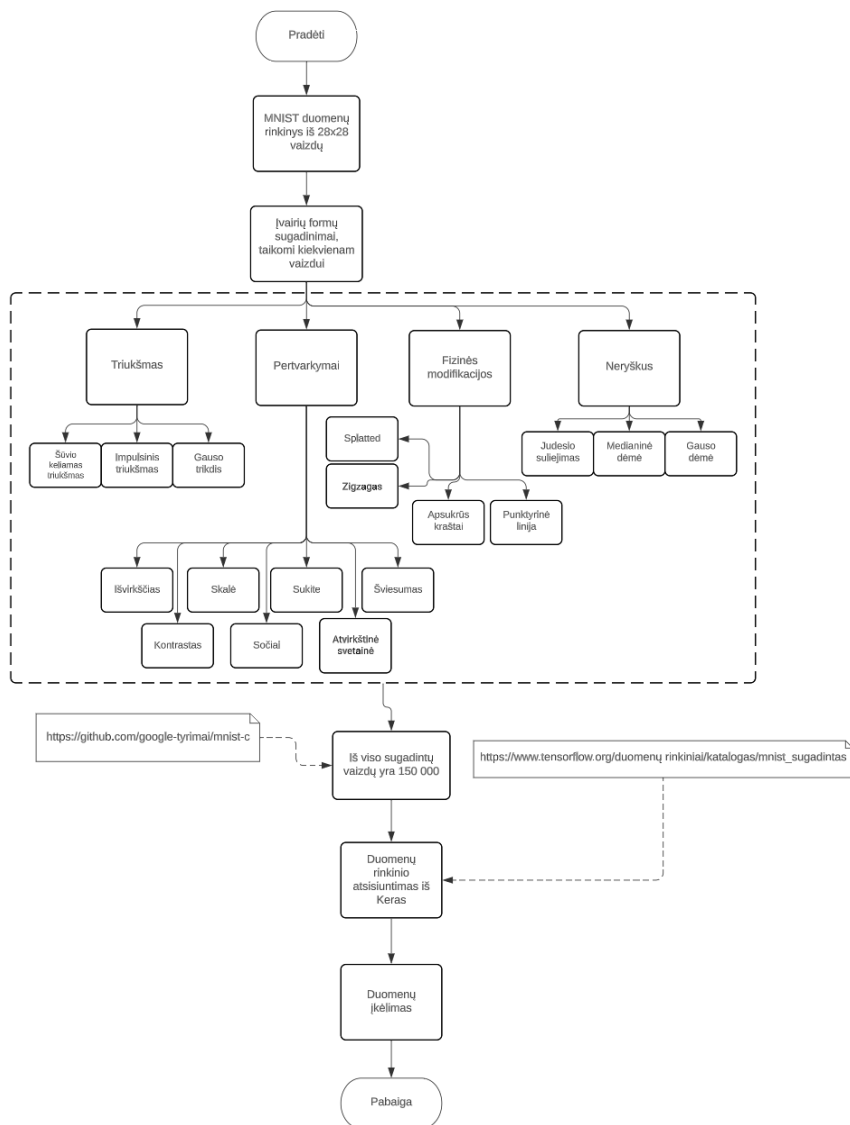
3 pav. Duomenų rinkinio *Fashion-MNIST* duomenų rinkimo vizualizacija

3.2.4 MNIST-C

Šį duomenų rinkinį sudaro 15 pažeidimų tipų, taikomų MNIST testų rinkiniui. Jo paskirtis – būti etalonu vertinant kompiuterinės regos modelių atsparumą ne pagal pasiskirstymą. MNIST-C duomenų rinkiniu siekiama atspindėti platų ir įvairų šiuolaikiniams modeliams būdingų gedimų spektrą. **4 pav.** pateikti anomalijų duomenų rinkinio pažeidimų tipai.

- Šūvio triukšmas: atsitiktinai iš vaizdo iškrenta keletas pikselių.
- Impulsinis triukšmas: atsitiktinai apverčiami kai kurie vaizdo pikseliai.
- Gauso triukšmas: į vaizdą įterpiamas Gauso triukšmas.
- Kvantavimo triukšmas: kvantavimas pikselių verčių iki nedidelio skaičiaus diskrečiųjų lygių.
- Kontrastas: pakeiskite vaizdo kontrastą.
- Elastinis iškraipymas: pritaikykite vaizdai elastingą iškraipymą.

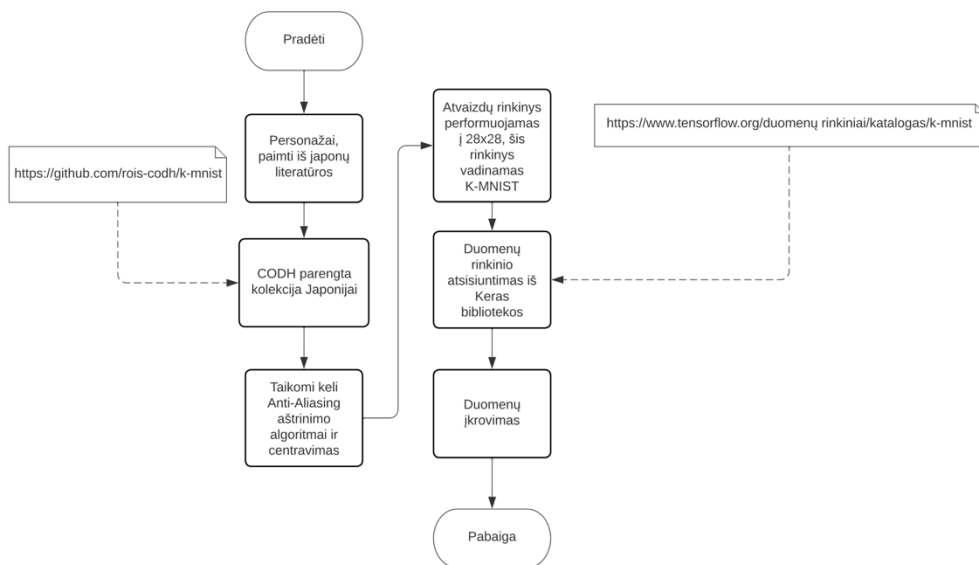
- Pikselizacija: pakeiskite vaizdo mastelį iki mažesnio dydžio, o tada padidinkite jį iki pradinio dydžio.



4 pav. Duomenų rinkinio MNIST-C duomenų rinkimo vizualizacija

3.2.5 K-MNIST

Mašinų mokymosi modeliams mokyti, ypač optinio ženklų atpažinimo srityje, naudojamas vaizdų rinkinys yra *Kuzushiji-MNIST* duomenų rinkinys (OCR). K-MNIST duomenų rinkinį sudaro 70 000 pilkos spalvos paveikslėlių, kurių kiekvienas vaizduoja vieną japonų hiraganos rašto ženklą. Kadangi kiekvienas paveikslėlis yra 28×28 pikselių kvadratas, kaip ir originalus MNIST duomenų rinkinys, jį galima naudoti su dabartiniais mašininio mokymosi modeliais ir sistemomis. Šviežių, dar neatrastų hiraganos rašto ženklų pakitimus galima rasti taikant išmokus modelius. Modelių našumui įvertinti galima taikyti tokius rodiklius kaip tikslumas, atšaukimas ir F1 balas, kaip parodyta **5 pav.**



5 pav. Duomenų rinkinio K-MNIST duomenų rinkimo vizualizacija

3.2.6 Nulinių žinių įrodymas

Taikydamos ZKP, organizacijos gali sukurti saugias ir privatumą išsaugančias anomalijų aptikimo sistemas, kurias gali audituoti ir tikrinti išorės šalys. ZKP

padeda aptikti anomalijas taikant statistinį metodą, pavyzdžiui, Z balų, išlaikant anonimiškumą.

$$Z = (X - \mu) / \sigma. \quad (1)$$

Z balas žymimas Z , o duomenų elementas – X . 1 lygtyje pateiktas duomenų rinkinio vidurkis yra duomenų rinkinio standartinis nuokrypis. Norint ZKP taikyti privatumą išsaugančiam anomalijų aptikimui, Z balas apskaičiuotas teisingai, neatskleidžiant jokios informacijos apie X , μ ir σ .

Įvairių klasifikavimo algoritmų, tokių kaip logistinė regresija, sprendimų medžiai, atsitiktiniai miškai, atraminių vektorių mašinos arba neuroniniai tinklai, naudojimas siekiant sukurti modelį, pagal kurį būtų galima numatyti anomalijas, kaip 2 lygtyje.

$$y = f(x); \quad (2)$$

čia $y = 1$ anomalijoms x ir $y = 0$, kai x yra normalus; x – įvesties požymių vektorius, o f – funkcija, kuri atvaizduoja įvestį į išvestį. Funkcija f gali būti išmokta taikant prižiūrimojo mokymosi algoritmą, kuris minimizuoja mokymo duomenų prognozavimo paklaidą..

3.2.7 Neprižiūrimojo mokymosi sistema

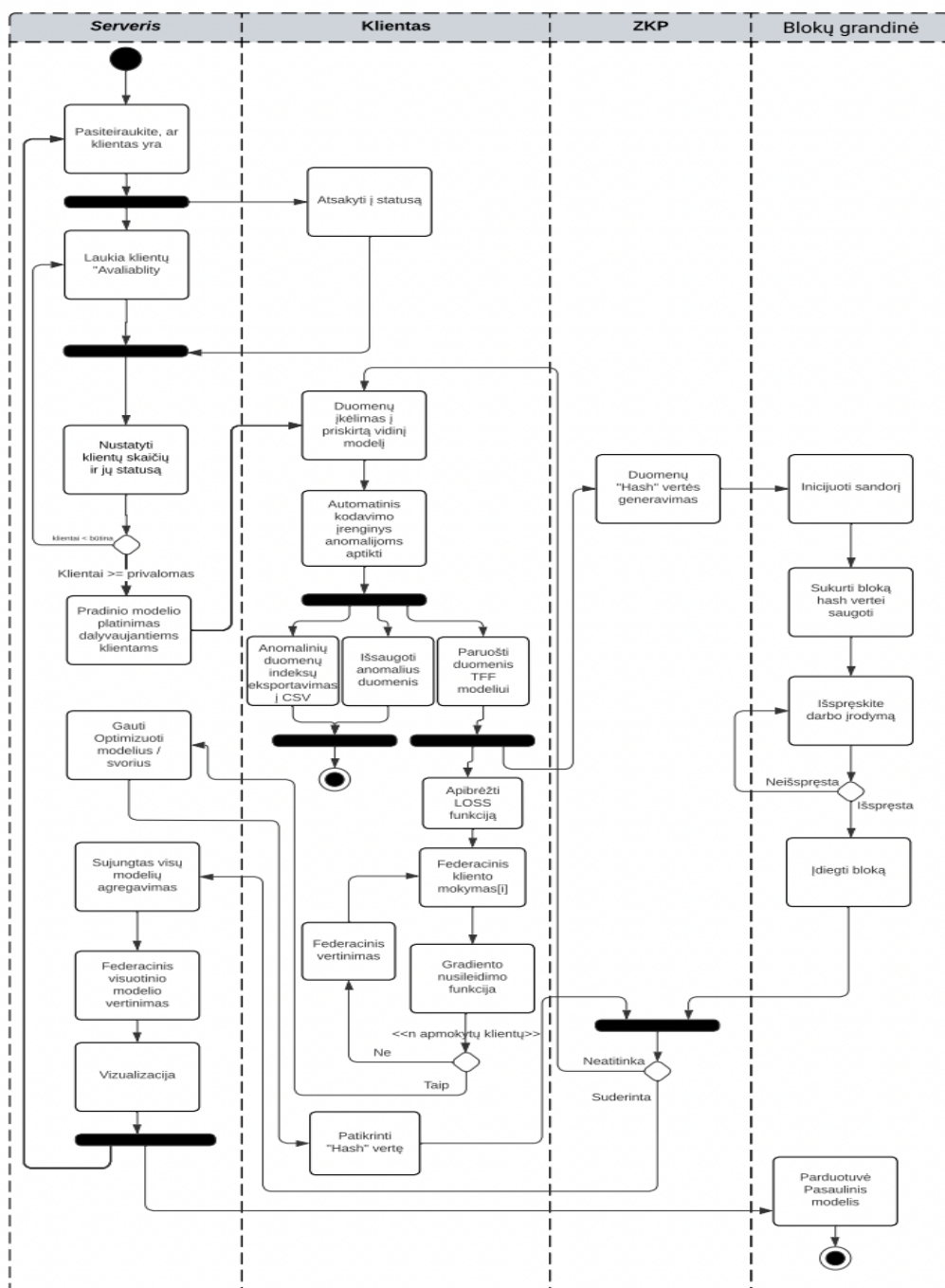
Tai mašininio mokymosi metodas, kai modelis mokomas iš nepažymėtų duomenų, t. y. nėra iš anksto nustatytų kategorijų ar klasių. Klasterizavimo algoritmai sugrupuoja panašius duomenų taškus, o asociacijų algoritmai nustato modelius ir ryšius tarp kintamųjų, kaip (3) ir (4) lygtyse toliau.

$$\text{Anomalijų balas} = (1 - \alpha) \times \text{vietinis anomalijų balas} + \alpha \times \text{visuotinis anomalijų balas} \quad (3)$$

čia α – svertinis koeficientas nuo 0 iki 1 ir nustato pusiausvyrą tarp vietinių ir visuotinių anomalijų balų.

$$\text{Vietinis anomalijų balas} = |X - \mu| / \sigma; \quad (4)$$

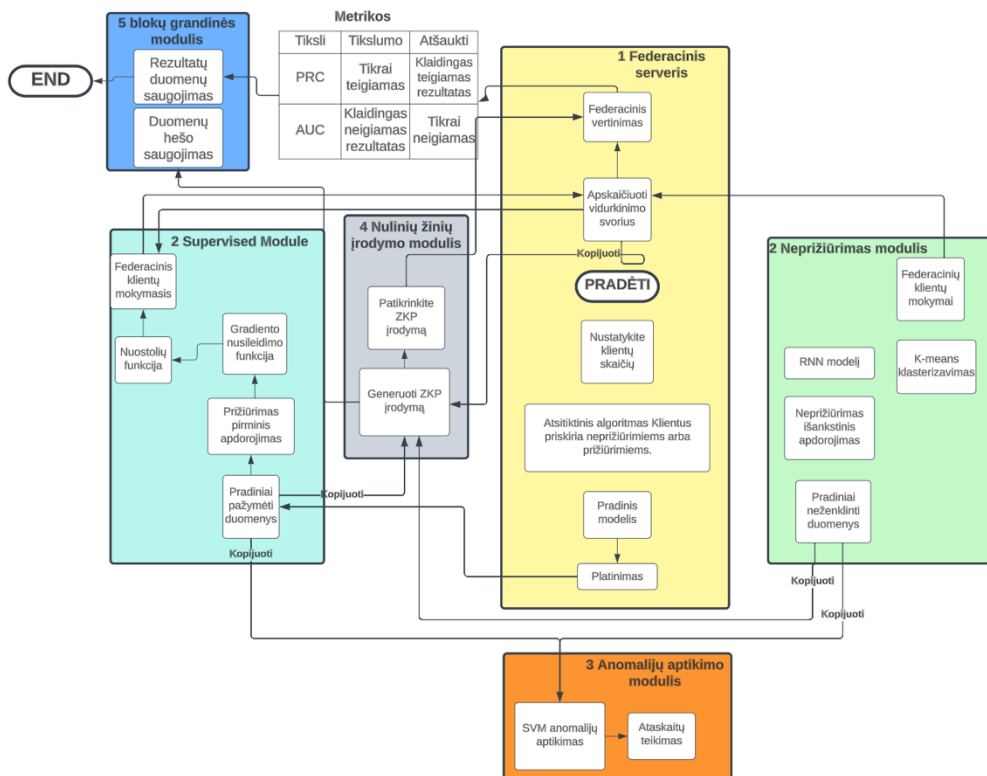
čia X – įvesties duomenų taškas, μ – vietinio duomenų pasiskirstymo vidurkis ir σ – vietinio duomenų pasiskirstymo standartinis nuokrypis. Federacinio mokymosi, blokų grandinės technologijos ir ZKP derinys gali užtikrinti saugų ir privatumą užtikrinantį nekontroliuojamąjį mokymąsi, kaip siūloma **6 pav.**



6 pav. Neprižiūrimojo mokymosi metodas taikant susietąjį mokymąsi, blokų grandinės technologiją ir ZKP.

3.3 Taikymo sistema

Anomalijų aptikimo taikant nulinių žinių įrodymus (ZKP), federacinę mokymąsi ir blokų grandinės technologiją sistema sujungia šių trijų technologijų privalumus, kad būtų sukurta saugi, privati ir decentralizuota sistema, skirta pažeidimams duomenų rinkiniuose nustatyti. Pirmyn nukreiptas procesas, svorio perdavimas pasauliniam serveriui, kad šis jį atnaujintų, ir grįžtamasis procesas yra trys pagrindiniai viso įgyvendinimo etapai. Visas šis procesas federaciniame mokyme vadinamas vienu komunikacijos ciklu. Po to modeliui apmokyti būtų taikomi ZKP metodai, siekiant užtikrinti, kad nebūtų atskleisti nė vienos šalies duomenys. ZKP taikomas skaičiavimams patvirtinti anomalijų aptikimo proceso metu neatskleidžiant jokių neskelbtinų duomenų. Išmokytas modelis naudojamas tikra laikams pažeidimams duomenų srautuose aptikti. Sistemoje taikomi ir prižiūrimieji, ir neprižiūrimieji metodai. Prižiūrimasis metodas leidžia kelioms šalims dirbti kartu nesidalijant duomenimis, nes pažymėti duomenys naudojami modeliui apmokyti taikant federacinę mokymąsi. Apmokytam modeliui saugoti naudojamas nekeičiamas ir skaidrus blokų grandinės tinklas, kaip parodyta 7 pav.



7 . Anomalijų aptikimo sistema taikant ZKP, FL ir blokų grandinės technologiją, prižiūrimąjį ir neprižiūrimąjį metodus

Nekontroliuojami klientai iš anksto apdoroja neženklintus duomenis, išgauna paslėptus požymius taikydami RNN modelį, klasterizuoja duomenis taikydami K vidurkių algoritmą ir apmoko savo pradinio modelio kopiją. Iš neženklintų duomenų gautų svorių maiša generuojama taikant ZKP modulį ir saugoma blokų grandinėje.

Serveris apskaičiuoja gautų svorių maišą, patikrina ją su anksčiau saugoma maišos funkcija ir apskaičiuoja svorių vidurkinimą. Tada šie vidutiniai svoriai siunčiami atgal, kad būtų pakeistas pradinis vietinis modelis, kuris po to iš naujo apmokytas pagal klientų sukurtus naujus svorius.

Anomalijų aptikimas yra labai svarbus uždavinys, kuris tampa vis sudėtingesnis dėl dinamiško šiuolaikinių duomenų pobūdžio, technologinės pažangos ir didelės bei nevienalytės duomenų apimties. Tradicinės anomalijų aptikimo sistemos neatitinka šiuolaikinių duomenų reikalavimų, pavyzdžiui, mastelio keitimo bei pritaikomumo, ir joms sunku veiksmingai apdoroti bei analizuoti didelius bei įvairius duomenų rinkinius. Šiems iššūkiams spręsti šitame skyriuje siūlomas unikalūs ir daugialypis metodas, kuriame derinamas federacinis mokymasis, nulinių žinių įrodymai ir blokų grandinės technologija. Integruota technologija turi didžiulę realią vertę ir gali pakeisti anomalijų aptikimo metodus. Siūlomas metodas užtikrina privatumą, vientisumą ir atskaitomybę, kartu padidindamas mastelio keitimą, pritaikomumą ir interpretuojamumą.

4 EKSPERIMENTINIAI TYRIMAI

4.1 Apžvalga

Šiame skyriuje apibrėžiame federacinį mokymosi modelį ir išmaniają sutartį, skirtą blokų grandinės tinklui, naudodami *Syft* ir *SmartPyme* bibliotekas. Modelis mokomas naudojant decentralizuotus duomenis, o gauti modelio svoriai užšifruojami taikant nulinių žinių įrodymo metodus. Tada šie užšifruoti svoriai saugiai išsaugomi išmaniojoje sutartyje, kad būtų galima efektyviai ir patikimai prognozuoti anomalijas. Užšifruoti modelio svoriai taikomi anomalijoms prognozuoti kaip proceso dalis. Prognozės iššifruojamos taikant nulinių žinių įrodymą, o rezultatai rodomi konsolėje.

4.2 Eksperimentinė sąranka. Ištekliai

Visiems bandymams virtualioje *Jupyter Notebook* debesijos aplinkoje buvo taikomi šie nustatymai, kaip **2 lentelėje** parodyta galiojimo trukmė.

2 lentelė. Eksperimentinės aplinkos sąranka

Techninė įranga	Konfigūracija
CPU	<i>Intel Celeron</i> procesorius @ 1,80 GHz
Standusis diskas	500 GB talpa
Atmintis	8 GB
GPU	<i>Intel ® HD Graphics</i>

Šiame projekte nuolat buvo naudojamos *PySyft* bibliotekos saugiam ir privačiam giluminiam mokymuisi. Duomenims iš anksto apdoroti ir vertinimo rodikliams pateikti dažniausiai naudojome *scikit-learn* įrankius. Vietiniai modeliai ir federacinio mokymosi modeliai buvo konstruojami naudojant *Pytorch*.

4.3. Prižiūravimo mokymosi metodas anomalijoms aptikti taikant nulinių žinių įrodymą kartu su federaciniu mokymusi ir blokų grandinės technologija

Anomalijų aptikimas yra svarbi problema daugelyje sričių, įskaitant finansų, kibernetinio saugumo ir defektų identifikavimo sritis. Taikant tradicinius anomalijų aptikimo metodus dažnai būtina centralizuota prieiga prie neskelbtinų duomenų, todėl kyla privatumo problemų. Vienas iš galimų metodų yra sutelktinis mokymasis, kuris leidžia bendrai mokytis modelius neatskleidžiant pirminių duomenų. Šio eksperimentinio projekto tikslas – ištirti federacinio mokymosi pritaikymą anomalijoms aptikti. Pagrindinis šio tyrimo tikslas – nustatyti federacinio mokymosi strategijos veiksmingumą ieškant anomalijų tam tikrame duomenų rinkinyje. Tyrimas apima SVM modelio anomalijoms nustatyti mokymą, vizualiųjų anomalijų atvaizdų kūrimą, duomenų parengimą federaciniam *TensorFlow* (TFF) modeliui ir federacinio mokymosi metodo efektyvumo vertinimą.

4.3.1 Eksperimento etapai

Duomenų rinkinio įkėlimas į atmintį: pirmasis žingsnis – įkelti duomenų rinkinį į atmintį. Šiame etape duomenų rinkinys nuskaitomas iš failo ir išsaugomas tinkamoje duomenų struktūroje, kad vėliau būtų galima jį apdoroti.

Prieš pateikiant patikslintus modelio parametrus į serverį, patikrinami 6 veiksmu sukurti ZKP, kad būtų patvirtinta, jog duomenys nebuvo suklastoti ar pakeisti per visą sutelktojo mokymosi procesą. Nustatomas mokymosi greitis 0,1, o nulinių žinių įrodymo patvirtinimas iš pradžių nustatomas kaip „initial_model“. Nustatomas 5 raundų skaičius, o nuostolių ir mokymosi rodikliai išsaugomi

atskiruose sąrašuose. Prieš pradėdant mokymo procesą, duomenys paverčiami *Panda DataFrame* „df“. Modelis mokomas naudojant mokymosi normą ir federacijos mokymo duomenis naudojant „federated_train“. Jei raundų skaičius yra du, triukšmo reikšmė yra vienas; priešingu atveju ji lygi nuliui. Įrodymas patvirtinamas naudojant „verify_proof“, kuris įvertina įrodymo galiojimą, atsižvelgiant į stulpelių reikšmes ir atsitiktines reikšmes. Jei įrodymas teisingas, mokymosi sparta padauginama iš 0,9, o nuostolio vertė apskaičiuojama naudojant „federated_eval“. Apskritimo skaičius ir nuostolio vertė spausdinami po to, kai nuostolio vertė ir mokymosi koeficientas pridedami prie atitinkamų sąrašų. Jei įrodymas negalioja, partija praleidžiama, kaip nurodyta 1 algoritme, o 3 lentelėje pateikiami visi rezultatai, kuriuos gavome įgyvendinę ZKP patvirtinimą. Kliento serverio atsakyme apie kiekvieną raundą pateikiami nuostolių skaičiai.

1 Algoritmas: ZKP patvirtinimas

```

1. modelis = initial_model
2. learning_rate = 0.1
3. number_of_rounds = 5
4. loss_values = []
5. learning_rates = []
6. df = pd.DataFrame(federated_train_data)
7. for round_num in range(0,number_of_rounds):
8.     modelis = federated_train(modelis, learning_rate, federated_train_data)
9.     jei
10.    number_of_rounds == 2:
11.        triukšmas = 1
12.    kitaip:
13.        triukšmas = 0
14.    jei
15.        verify_proof(df, column_values, proof, random_values, noise):
16.            print("ZKP VALIDITY PASSED, CLIENT -> Seerver IS SECURE")
17.            learning_rate = learning_rate * 0,9
18.            loss = federated_eval(modelis, federated_train_data)
19.            loss_values.append(loss)
20.            learning_rates.append(learning_rate)
21.            spausdinti("round {}, loss={}".format(round_num, loss))
22.        kitaip:
23.            print("ZKP VALIDITY CHECK FAILED, praleidžiama ši partija.
```

3 lentelė. ZKP patvirtinimas

ZKP validumas	Klientas	Raundas	Nuostoliai
PASSED	Serveris yra saugus	0	20.691389083862305
PASSED	Serveris yra saugus	1	19.16118049621582

PASSED	Serveris yra saugus	2	17.984769821166992
PASSED	Serveris yra saugus	3	17.064706802368164
PASSED	Serveris yra saugus	4	16.326141357421875

Jei ZKP galioja, kliento įrenginiai gali saugiai perduoti atnaujintus modelio parametrus į serverį. Siekiant išsaugoti duomenų privatumą ir vientisumą, ši ryšio procedūra turėtų būti šifruojama. Sviurių federacinį vertinimą apskaičiuoja serveris, gavęs atnaujintus modelio parametrus iš kliento įrenginių, serveris atlieka federacinį vertinimą apibendrinamas kiekvieno kliento svorius. Pagal taikomą federacinio mokymosi sistemą šis apibendrinimas gali būti atliekamas taikant tokius metodus kaip federacinis vidurkinimas arba svertinis vidurkinimas.

Priešingu atveju mokymo duomenų partiją praleiskite. Jei tikrinimo etape nustatoma, kad ZKP yra klaidingi, tai reiškia, kad jais buvo manipuluojama arba elgiamasi kenkėjiškai. Tokiomis aplinkybėmis serveris praleidžia susijusią kliento mokymo duomenų partiją ir pereina prie kitos partijos.

Galiausiai rezultatus įrašykite į blokų grandinę. Siekiant skaidrumo, nekeičiamumo ir galimybės atlikti auditą, federacinio vertinimo rezultatus galima saugoti blokų grandinėje. Rezultatų saugojimas blokų grandinėje užtikrina, kad jų negalima pakeisti ar jais manipuluoti po fakto, todėl pateikiamas patikrinamas federacijos mokymosi proceso ir jo rezultatų įrašas.

4 lentelė. Blokų grandinės patikrinami įrašai

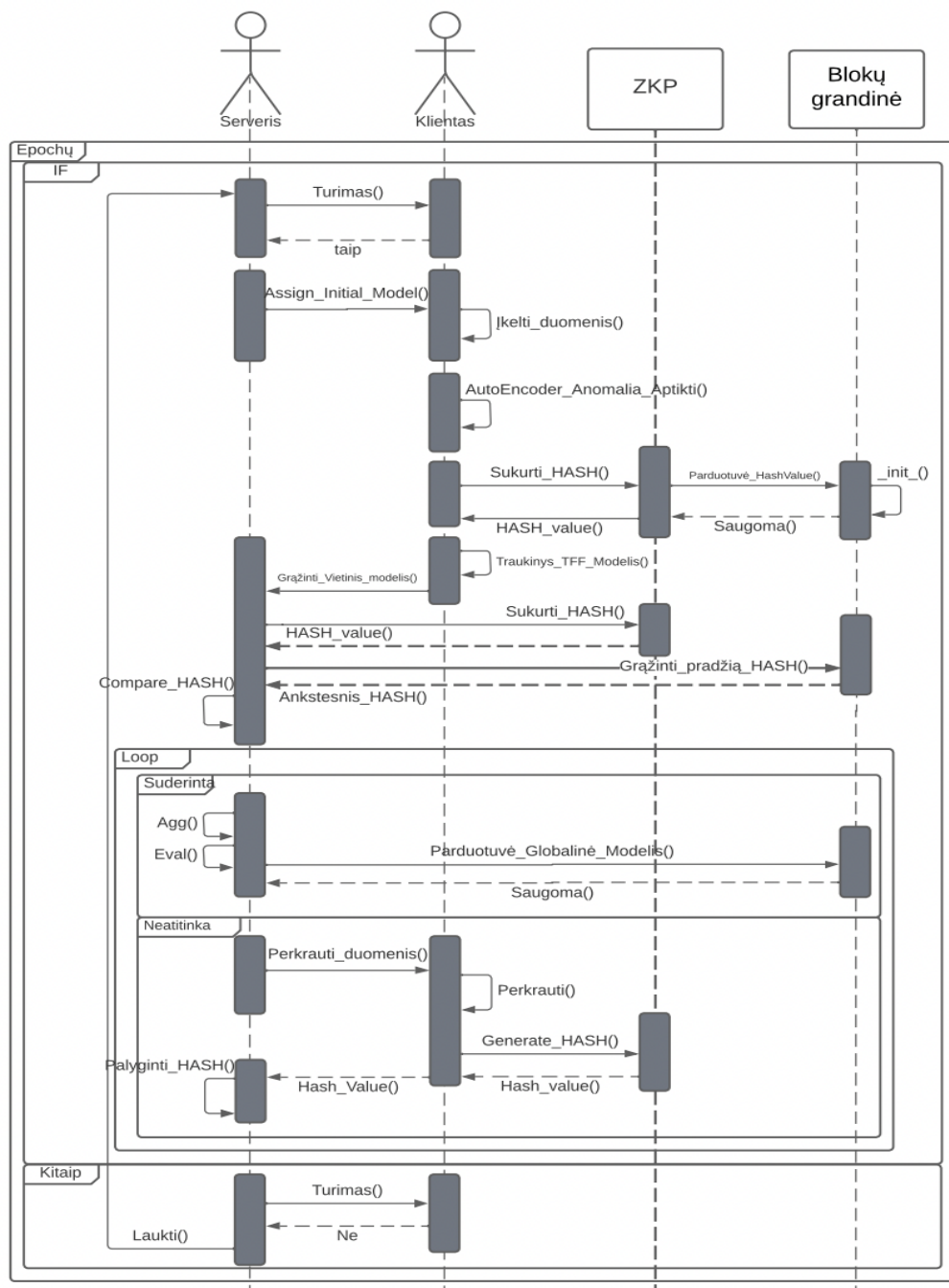
Indek- sas	Laiko žyma	Duome- nys	Įro- dy- mas	Maišos reikšmė
2	167984266 0.113144	5	3529 3	[d2b3d509682326c7d968f49beed8 5ad049503752a789fa69345a0e493 2002929]

algoritme pateikiamas prižiūrimojo mokymosi metodas, kuriame taikomi ZKP ir FL kartu su blokų grandinės technologija. Prižiūrimojo mokymosi metodo,

kuriame derinami ZKP, FL ir blokų grandinės technologija, kūrimo procesai pateikti **8 pav.** Šis metodas leidžia mums apmokyti savo modelį apsaugant duomenų konfidencialumą ir privatumą.

2 algoritmas: prižiūrimojo mokymosi metodas taikant ZKP ir FL kartu su blokų grandinės technologija

1. Skaityti() duomenų rinkinį iš failo "path/to/dataset.csv".
 2. Atlikti SVM anomalijų aptikimą.
 3. SVM modelio mokymas() naudojant "duomenų rinkinį".
 4. Nustatyti() anomalijas "duomenų rinkinyje".
 5. Išsaugokite() anomalinius duomenis kaip "anomalous_data.csv".
 6. Įkelti() duomenis iš "anomalous_data.csv".
 7. Generuoti() vaizdinius vaizdus pagal "anomalous_data".
 8. Išankstinis apdorojimas() "anomalous_data", skirtas TFF modeliui.
 9. Sukurti() ZKP iš anksto apdorotiems duomenims.
 10. Apibrėžti() susieto mokymosi modelio nuostolių funkciją.
 11. Kiekvienam klientui:
 - a) siųsti iš anksto apdorotus duomenis ir atitinkamą ZKP klientui;
 - b) atlikti vietinį mokymą kliento įrenginyje;
 - c) siųsti atnaujintą modelį į serverį.
 12. Kiekvienam klientui:
 - a) gaukite atnaujintą modelį iš serverio;
 - b) įvertinkite modelio veikimą naudodami vietos duomenis;
 - c) patikrinkite ZKP vientisumą;
 - d) jei (ZKP == galioja), vertinimo rezultatus nusiųskite į serverį;
 - e) priešingu atveju praleiskite mokymo duomenis (partiją).
 13. Apibendrinkite() iš kiekvieno kliento gautus modelio svorius.
 14. Saugokite() suvestinį modelį serveryje.
 15. Saugokite apibendrintus modelio ir vertinimo rezultatus blokų grandinėje.
-



8 pav. Prižiūrimojo mokymosi metodas taikant ZKP ir FL kartu su bloky grandinės technologija

4.3.2 Rezultatai

Savo darbe norimiems rezultatams pasiekti naudojome tris skirtingus duomenų rinkinius. Pirmasis duomenų rinkinys buvo *Fashion*-MNIST duomenų rinkinys, kurį sudaro įvairių drabužių atvaizdai. Antrasis duomenų rinkinys buvo MNIST duomenų rinkinys, kurį sudaro duomenys, susiję su įvairiais daiktų interneto įrenginiais, ir galiausiai E-MNIST duomenų rinkinys, kuris iš esmės yra MNIST duomenų rinkinio išplėtimas. Anomalijoms duomenyse aptikti dirbome prižiūrimojo mokymosi metodu, taikydami nulinio žinojimo įrodymą kartu su federaciniu mokymusi ir blokų grandinės technologija. Mūsų tikslas buvo užtikrinti, kad neskelbtini duomenys būtų privati ir saugūs, tačiau kartu būtų galima tiksliai aptikti anomalijas. Naudodami šį unikalų technologijų ir duomenų rinkinių derinį, sugebėjome pasiekti anomalijų aptikimo tikslumą trijuose skirtinguose duomenų rinkiniuose, kad galėtume pateikti daugiau rezultatų pagal *Fashion*-MNIST, MNIST, E-MNIST duomenų rinkinius.

4.3.2.1 Mados ir mados duomenų rinkinys



9 pav. *Fashion*-MNIST duomenų rinkinyje aptiktos anomalijos

Fashion-MNIST duomenų rinkinio rezultatai:

1. Pradinio modelio bandymo nuostoliai = 22,8 %.
2. Pradinio modelio bandymo tikslumas = 77,20 %.
3. Apmokyto modelio bandymo nuostoliai = 17,22 %.
4. Apmokyto modelio bandymo tikslumas 82,78 %.

Pradinio modelio bandymo nuostoliai yra svarbus parametras, kuriuo vertinamas skirtumas tarp numatytų ir faktinių testo duomenų verčių pradiniam modelio mokymo etape, kaip parodyta **10 pav.** Tiksliai prognozuotų testo duomenų procentinė dalis pradiniam modelio mokymo etape matuojama `initial_model` bandymo tikslumu. Šis statistinis rodiklis yra svarbus, nes parodo modelio gebėjimą patikimai numatyti duomenis, o tai labai svarbu, kad modelis būtų naudingas praktiniam taikymui. Kai modelis jau apmokytas, apmokyto modelio bandymo nuostoliai įvertina skirtumą tarp prognozuotų ir faktinių testo duomenų reikšmių. Po to, kai modelis buvo apmokytas, apmokyto modelio bandymo tikslumas parodo teisingai prognozuotų testinių duomenų procentinę dalį. Po to, kai modelis buvo apmokytas, apmokyto modelio bandymo tikslumas rodo tinkamai prognozuotų

testinių duomenų procentinę dalį. Ši statistika yra svarbi, nes parodo modelio gebėjimą patikimai prognozuoti duomenis, o tai būtina praktiniam taikymui. Didesnis bandymo tikslumo skaičius rodo, kad modelis buvo tinkamai apmokytas ir gali patikimai prognozuoti bandymo duomenis. Iš pateiktų teiginių aišku, kad apmokyto modelio našumas, palyginti su pradiniu modeliu, padidėja bandymo nuostolių (sumažėjo nuo 22,8 % iki 17,22 %) ir bandymo tikslumo (padidėjo nuo 77,20 % iki 82,78 %) atžvilgiu. Šie rodikliai rodo, kaip gerai veikia modelis ir kaip veiksmingai mokymo procedūra pagerino jo prognozavimo įgūdžius.

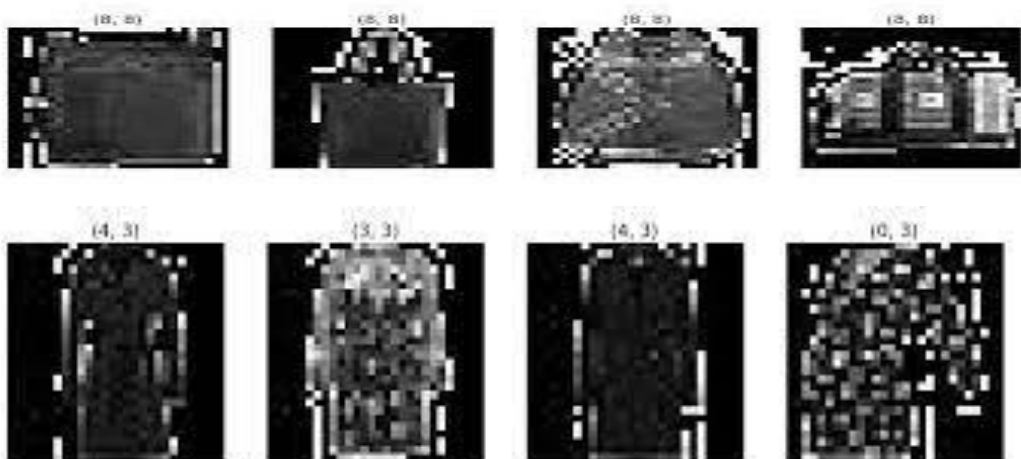
Išvestis: išvestį galima vizualizuoti 3 būdais.

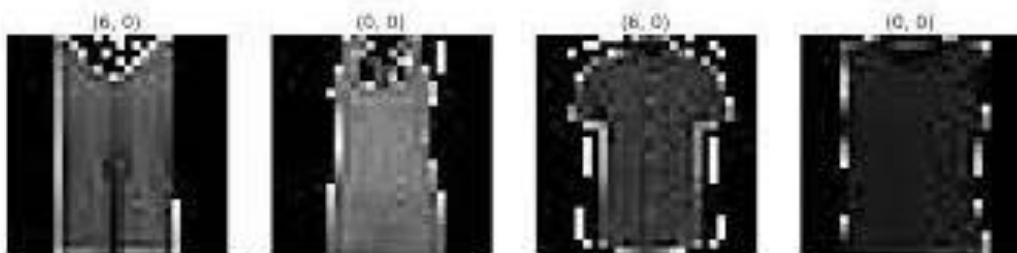
1. CSV failo generavimas.
2. Anomalių vaizdų indeksai (5 lentelėje pateikiamos visos indeksų vertės, kurios aptiktos kaip anomalijos *Fashion-MNIST* duomenų rinkinyje).
3. Vizualizacija

5 lentelė. Anomalių indeksų reikšmės *Fashion-MNIST* duomenų rinkinyje

[1 2 3 4 5 7 9 10 11 12 14 16 17 18 19 20 21 22
23 24 25 26 27 29 30 31 32 33 34 37 38 39 41 42 43 44
45 46 47 48 51 53 56 57 59 60 61 62 63 64 65 66 67 68
69 70 71 72 74 79 80 82 83 84 87 88 89 90 91 92 93 95
96 97 98 99 100 101 102 103 104 105 106 108 109 110 111 112 113
114
115 116 117 118 119 120 121 122 123 124 125 126 127 129 131 132
133 135
136 137 138 140 141 142 145 146 149 151 153 156 157 158 159 160
162 165
166 167 169 170 171 172 173 174 175 176 177 178 179 180 184 185
187 188
189 190 191 192 193 194 195 196 198 200 201 202 203 204 205 206
208 210
211 212 214 215 217 218 220 221 222 224 226 227 228 229 230 231
232 233
236 237 240 242 243 244 245 247 249 250 252 253 254 255 256 257
258 259
260 261 263 264 266 268 269 270 272 275 276 278 279 280 281 283
284 285
287 288 289 291 292 293 294 295 297 298 300 301 303 304 305 306
308 309
310 311 312 313 314 315 318 321 322 324 326 327 329 330 331 332
333 334
336 337 340 341 342 344 346 347 348 349 351 352 353 354 355 356
358 359
360 363 365 367 369 372 373 374 375 377 380 381 382 384 385 386

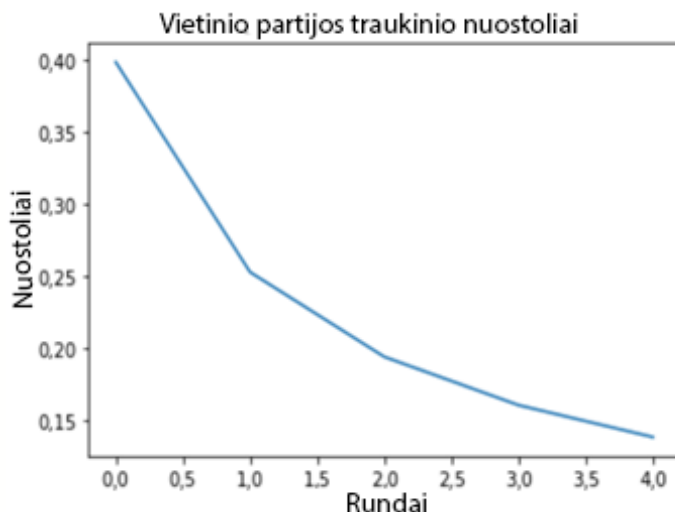
387 388
 391 392 393 394 395 396 397 398 399 401 402 403 404 405 406 408
 409 410
 411 412 413 414 416 417 418 419 420 421 422 423 425 426 429 430
 431 432
 433 434 435 436 437 439 440 441 442 444 445 446 447 450 451 454
 455 456
 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 474
 476 477
 479 480 483 484 485 486 487 489 490 491 492 493 494 495 496 497
 498 499
 501 503 504 505 506 507 510 511 512 513 514 515 516 517 518 519
 522 523
 524 525 526 527 528 530 531 532 533 535 536 538 539]



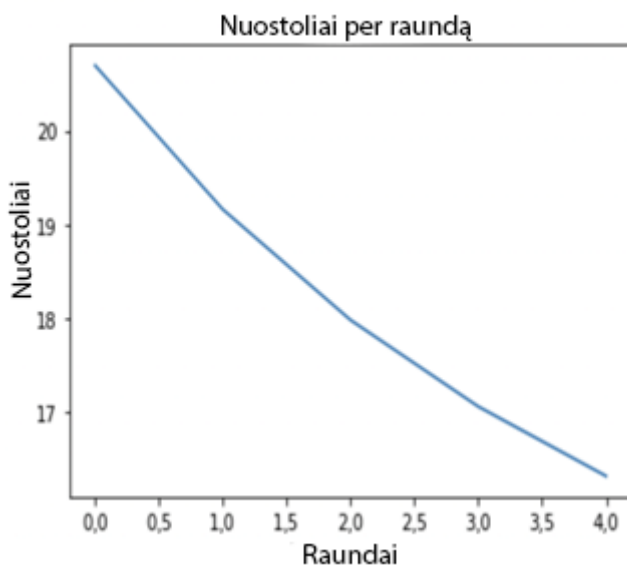


10 pav. Fashion-MNIST duomenų bazės vizualizacijos anomalijos

11 pav. pateikiami vietiniu būdu apmokyto modelio nuostoliai vietinėje partijoje, **12 pav.** – vietiniu būdu apmokyto modelio nuostoliai kiekviename raunde. Pradiniai modelio nuostoliai priklauso nuo pradinio modelio kokybės ir jam apmokytų naudotų duomenų. Vietos lygmeniu apmokyto modelių nuostoliai skirsis pagal vietos duomenų kokybę ir mokymo procesą.



11 pav. Vietos lygmeniu apmokyto modelio nuostoliai vietinėje partijoje



12 pav. Vietos lygmeniu apmokyto modelio nuostoliai per raundą

4.4.1 Fashion-MNIST duomenų rinkinys

4.4.1.1 Parametrai ir sluoksniai

sequential_1 nuoseklusis modelis, t. y. kiekvienas sluoksnis yra sujungtas su ankstesniu, vienas po kito. Pirmasis sluoksnis yra „suplokštinimo“ sluoksnis, kuris įvesties duomenis paverčia vienmačiu masyvu. Antrasis sluoksnis yra „tankus“ sluoksnis su 128 neuronais, o tai reiškia, kad jis turi 128 mokomuosius parametrus. Trečiasis sluoksnis taip pat yra „tankus“ sluoksnis su 10 neuronų, kurie atitinka 10 galimų išvesties klasių. Šis modelis iš viso turi 100 608 treniruojamus antrojo sluoksnio parametrus ir 1 290 treniruojamų trečiojo sluoksnio parametrų. Toliau **6 lentelėje** paminėtos visos reikšmės, kurias gavome iš paleistų sluoksnių.

6 lentelė. Fashion-MNIST duomenų rinkinio apmokytos ir neapmokytos parametrų reikšmės

Iš viso parametrų	101 898
Treniruojamieji parametrai	101 898
Netreniruojamieji parametrai	0

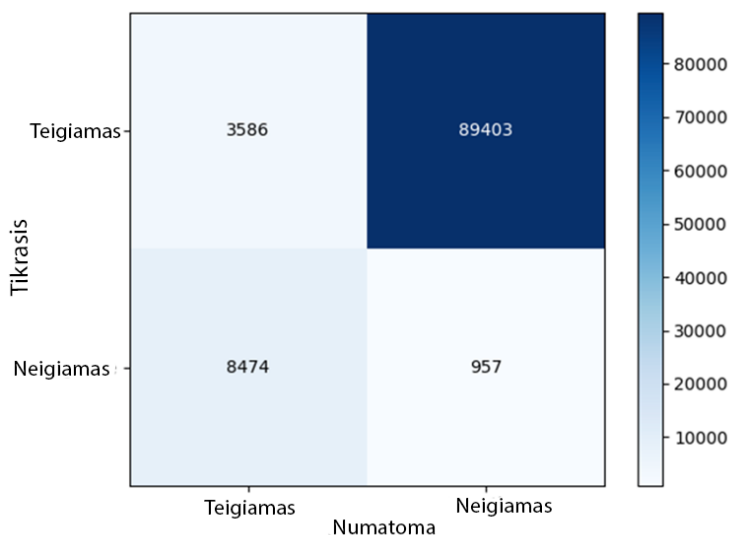
4.4.2 Fashion-MNIST Rezultatai (tęsinys)

7 lentelė. Neprižiūrimų Fashion-MNIST nuostolių sumažinimas pagal epochą

Epocha	Nuostoliai	Patvirtinimo nuostoliai
1	0,3869	0,3261
2	0,3137	0,3080
3	0,3010	0,2984
4	0,2940	0,2933
5	0,2901	0,2905
6	0,2879	0,2887
7	0,2865	0,2878
8	0,2856	0,2869
9	0,2850	0,2865
10	0,2846	0,2861

7 lentelėje pateikiamos modelio mokymo proceso metu gautos mokymo ir patvirtinimo nuostolių vertės pagal epochas. Kiekviena epocha reiškia visą mokymo duomenų rinkinio iteraciją. Nuostolių stulpelyje nurodomas vidutinis nuostolis (arba paklaida) tarp prognozuojamų ir tikrųjų verčių mokymo duomenų rinkinyje, o patvirtinimo nuostolių stulpelyje nurodomas vidutinis nuostolis atskirame patvirtinimo duomenų rinkinyje, kuris padeda įvertinti modelio apibendrinimo efektyvumą. Modelio mokymo nuostoliai yra 0,3869, o patvirtinimo nuostoliai – 0,3261 pirmojoje epochoje. Tęsiant mokymą, tiek mokymo, tiek patvirtinimo nuostoliai laipsniškai mažėjo. Penktojoje epochoje modelio mokymo nuostoliai buvo 0,2901, o patvirtinimo nuostoliai – 0,2905. Šis modelis išliko iki vienuoliktosios epochos, kai mokymo nuostoliai buvo 0,2846, o patvirtinimo nuostoliai – 0,2861. Mažėjančios mokymo ir patvirtinimo nuostolių reikšmės didėjant epochoms rodo, kad modelio veikimas gerėja su kiekviena iteracija. Mažėjantys mokymo nuostoliai reiškia, kad modelis tapo tikslesnis prognozuojant mokymo duomenis, o mažėjantys patvirtinimo nuostoliai rodo, kad pagerėjo modelio apibendrinimo gebėjimas nežinomiems duomenims. Didelis tikslumo rezultatas rodo, kad modelio klaidingai teigiamų rezultatų lygis yra mažas, todėl sumažėja galimybė klaidingai suklasifikuoti neigiamus atvejus kaip teigiamus. Atšaukimo rodiklis, dar vadinamas jautrumu arba tikrojo teigiamojo rezultato rodikliu, nuolat viršija 84 % ir siekia net 97 %. Tiksliai numatytų teigiamųjų atvejų (tikrieji teigiamieji atvejai) ir visų faktinių teigiamųjų atvejų (tikrieji teigiamieji atvejai + klaidingai neigiamieji atvejai) santykis matuojamas atšaukimu. Didelis atšaukimo rezultatas rodo, kad modelis veiksmingai aptinka teigiamuosius atvejus, kartu sumažindamas klaidingų neigiamųjų atvejų skaičių. AUC balas, pagal kurį vertinamas bendras modelio veikimas esant skirtingoms klasifikavimo riboms, išlieka artimas vienetui, o tai rodo puikų teigiamųjų ir neigiamųjų atvejų atskyrimą. Panašiai ir PRC rodiklis išlieka nuolat didelis, o tai rodo, kad modelis sugeba

pasiekti didelį tikslumą ir atšaukimą, kartu išlaikydamas nuoseklumą. **8 lentelėje** pateiktos painiavos matricos reikšmės, o **9 lentelėje** iliustruojami vertinimo rodiklių rezultatai rodo puikų modelio veikimą. Per daugybę epochų jis nuolat gauna gerus tikslumo, atšaukimo, AUC ir PRC balus. Šie rezultatai rodo, kaip matote **13 pav.**, kad modelis sugeba tinkamai klasifikuoti įvykius, sumažinti klaidingai teigiamus ir klaidingai neigiamus rezultatus ir pasiekti didelį bendro našumo lygį.



13 pav. Prognozuojamas tikslumo rezultatas *Fashion*-MNIST duomenų rinkinio painiavos matrica

Šie rezultatai rodo, kad mokymo metodas padėjo sumažinti modelio nuostolius ir pagerino prognozavimo įgūdžius. Nuolatinis mokymo ir patvirtinimo nuostolių mažėjimas reiškia, kad modelis sėkmingai išmoko ir užfiksavo svarbius duomenų rinkinio modelius ir ryšius, todėl, didėjant mokymo epochų skaičiui, jo rezultatai gerėjo.

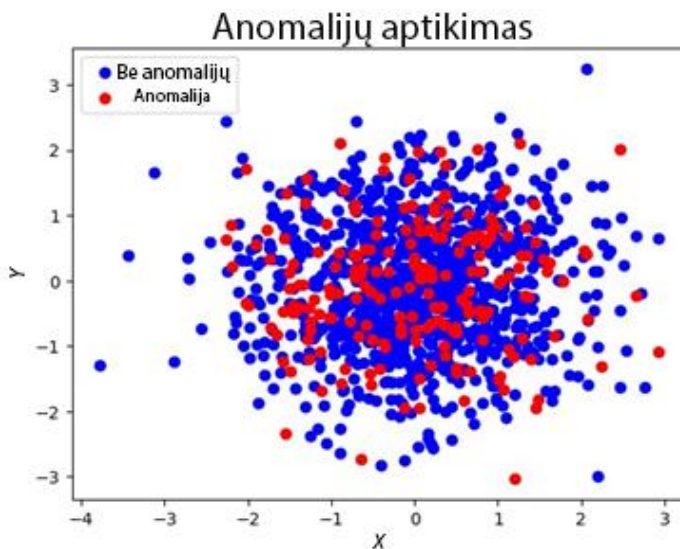
8 lentelė. Apmokytų epochų reikšmė pagal *Fashion*-MNIST duomenų rinkinio painiavos matricą

Epocha	Nuostoliai	Tikrieji teigiami	Klaidingai teigiami	Tikrieji neigiami	Klaidingai neigiami
1	0	0,4895	15 750	2388	177 612
2	1	0,409	8235	1056	88 944
3	2	0,383	8300	959	89 041
4	3	0,3853	8397	1041	88 959
5	4	0,3586	8474	957	89 043

9 lentelė. Apmokytų epochų reikšmė Fashion-MNIST duomenų rinkinio vertinimo rodikliai.

Dvejjetainis tikslumas	Tikslumas	Atšaukimas	AUC	PRC
0,99	0,96	0,93	1	0,98
0,99	0,97	0,95	1	0,99
0,99	0,97	0,96	1	0,99
0,99	0,97	0,97	1	0,99
0,99	0,98	0,97	1	1

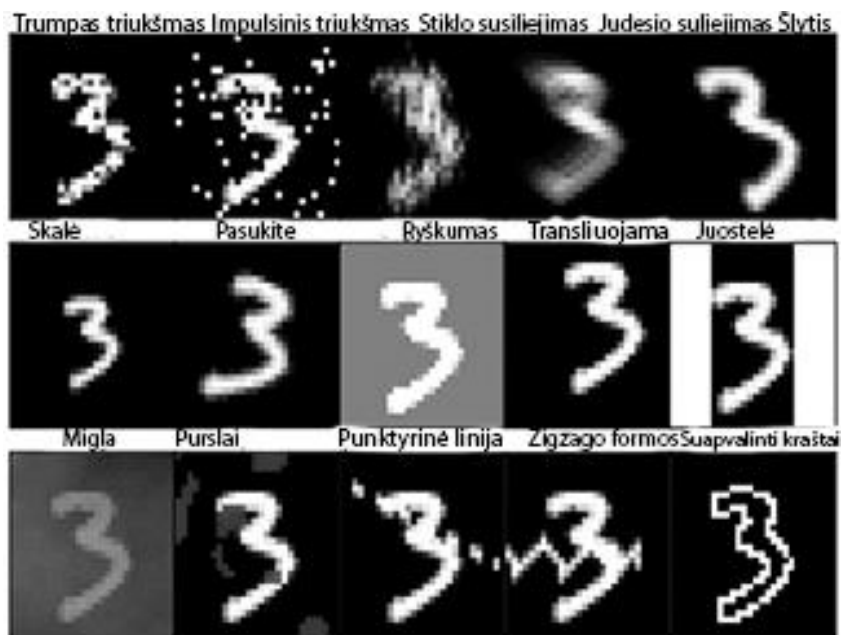
Mokymo proceso metu modelis pasižymi dideliu tikslumu, kuris siekia 98 % tiek mokymo, tiek patvirtinimo duomenų rinkiniuose. Vadinas, modelio prognozės tiksliai atitinka faktines etiketes, o tai įrodo jo gebėjimą tinkamai nustatyti įvykius. Modelio tikslumas nuolat didelis ir svyruoja nuo 94 % iki 98 %. Tiksliai nuspėtų teigiamųjų atvejų (tikrieji teigiamieji atvejai) ir visų numatytų teigiamųjų atvejų (tikrieji teigiamieji atvejai + klaidingai teigiami atvejai) santykis parodomas tokiu tikslumu, kaip 14 pav. pateiktas anomalijų aptikimo grafikas.



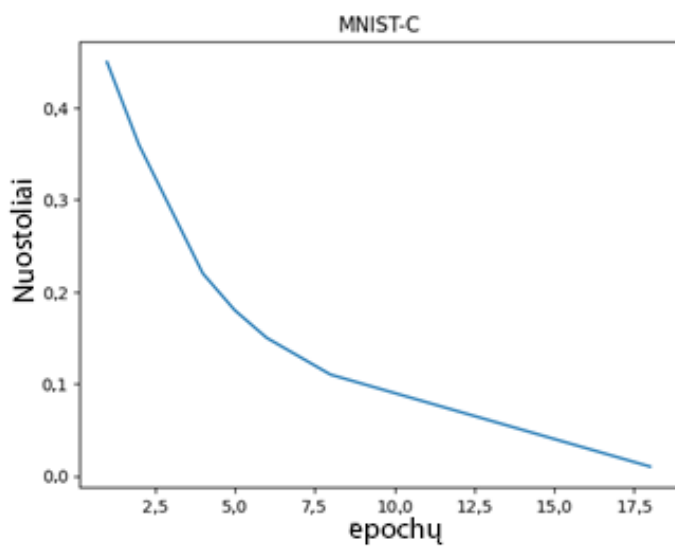
14 pav. Aptiktos anomalijos prognozuojamas grafikas *Fashion*-MNIST duomenų rinkinyje

4.4.3 MNIST-C duomenų rinkinys

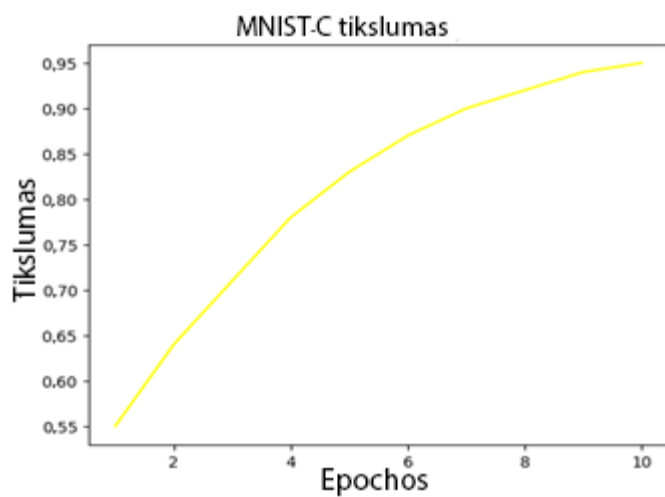
Duomenų rinkinys buvo pasirinktas siekiant išsamiai įvertinti modelio patikimumą vaizdo programose. Skirtingai nei įprastų vertinimo duomenų rinkinių, kuriuos daugiausia sudaro švarūs vaizdai, MNIST-C duomenų rinkinio privalumas yra tas, kad į jį dažnai įtraukiami realiomis aplinkybėmis pasitaikantys pažeidimai ir trikdžiai, todėl galima realiau įvertinti modelio veikimą. Pateikus modelį MNIST-C duomenų rinkiniui, buvo įvertintas modelio gebėjimas susidoroti su įvairių tipų ir lygių pažeidimais, tokiais kaip neryškumas, triukšmas, ryškumo pokyčiai ir sukimas. Šis išsamus pažeidimų tyrimas padėjo visapusiškai pažinti modelio apibendrinimo gebėjimus ir atskleidė galimus modelio konstrukcijos ar mokymo metodo trūkumus. Nustačius šiuos trūkumus, pastangas galima nukreipti į modelio patikimumo didinimą realiose kompiuterinės regos programose. Be to, naudojant MNIST-C duomenų rinkinį buvo galima atlikti lyginamąją analizę, modelį lyginant su kitais pažangiausiais kompiuterinės regos modeliais pagal tą patį lyginamąjį kriterijų. Tai leido objektyviai įvertinti modelio našumą, palyginti su konkurentais, ir gauti svarbios informacijos modelio parinkimui bei būsimiems pakeitimams. Siekdami patikrinti modelio tikslumą, dirbome su įvairiomis anomalijomis, kurios pateiktos 15, 16 ir 17 pav.



15 pav. Anomalijų aptikimas pagal kelis anomalius modelio elgsenos būdus



16 pav. Netroliuojamos partijos nuostoliai naudojant MNIST-C duomenų rinkinį

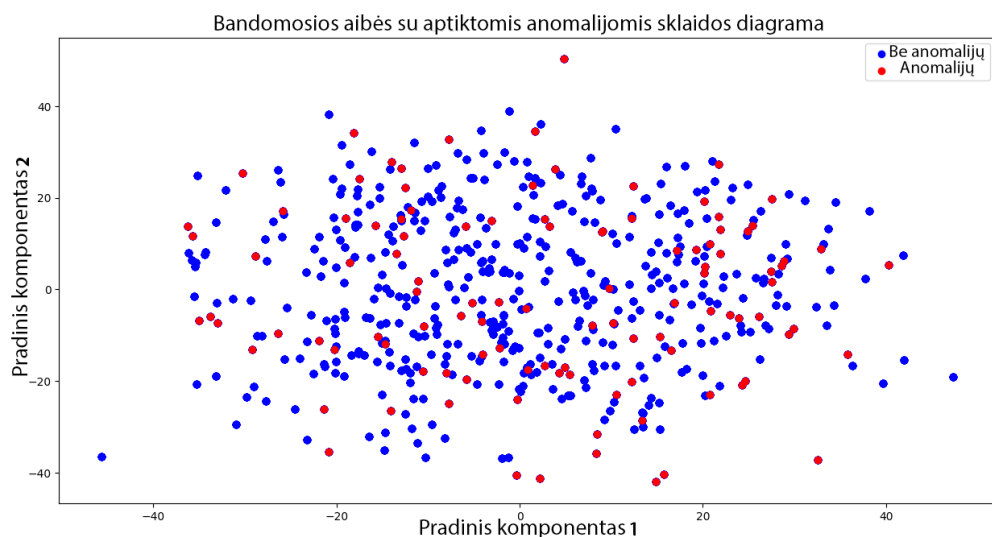


17 pav. Netroliuojamos partijos tikslumas naudojant MNIST-C duomenų rinkinį

10 lentelė. MNIST-C duomenų rinkinio prognozuojamos vertinimo metrikos

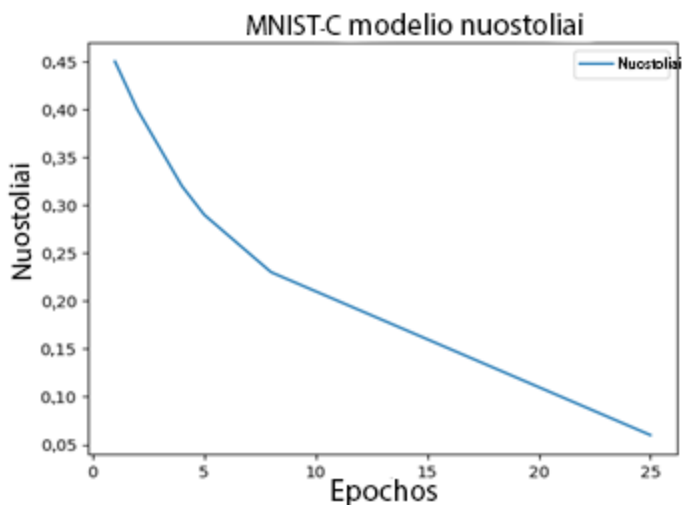
Tikslumas	0,95
Tikrųjų teigiamų rezultatų tikslumas	0,85
Atšaukimas	0,8
F1 rezultatas	0,777777777777778
Nuostoliai	0,03

Modelio tikslumas buvo 0,95, t. y. jis teisingai prognozavo 95 % situacijų. Toks didelis tikslumo rezultatas rodo, kad modelis gali iš esmės tiksliai prognozuoti. Be to, 0,85 tikslumas reiškia, kad modelis teisingai numatė 85 % visų teigiamų prognozių. Vadinasi, modelio klaidingai teigiamų prognozių lygis yra žemas, o tai įrodo jo gebėjimą pakankamai tiksliai nustatyti tikras teigiamas situacijas, kaip pateikta **10 lentelėje**. Be to, esant 0,8 atšaukimo rodikliui, modelis teisingai nustatė 80 % tikrųjų teigiamųjų atvejų. Nors tai rodo, kad duomenų rinkinyje pakankamai gerai pavyksta aptikti teigiamosius atvejus, vis dar yra galimybių tobulinti visų teigiamų situacijų fiksavimą. F1 balas sujungia tikslumą ir atšaukimą, kad būtų galima teisingai įvertinti modelio našumą. Aprašomame scenarijuje nustatytas F1 balas yra 0,7778. Šioje statistikoje atsižvelgiama ir į tikslumo, ir į atšaukimo vertes, taip pat pateikiamas išsamus modelio veiksmingumo įvertinimas. Apibendrinant galima teigti, kad modelis veikia gerai, jo tikslumo įvertinimai yra puikūs. Tačiau gerokai mažesnis atšaukimo rezultatas rodo, kad kai kurie teigiamieji atvejai galėjo būti nepastebėti. Norint įvertinti, ar modelio efektyvumas yra pakankamas konkrečiai užduočiai atlikti, labai svarbu išanalizuoti konkrečius problemos srities poreikius ir kontekstą. 0,03 nuostolių balas rodo, kad modelis pateikė gana tikslias prognozes. Nuostolių rodiklis apskaičiuoja skirtumą tarp tikėtinųjų ir tikrųjų verčių. Mažesnis nuostolių skaičius rodo, kad modelio prognozės yra tikslesnės už tikrąsias duomenų rinkinio reikšmes. Nuostolių skaičius 0,03 reiškia, kad modelio prognozės nuo tikrųjų verčių vidutiniškai nukrypsta nedaug. Vadinasi, modelis labai gerai įvaldė pagrindinius duomenų dėsningumus ir ryšius. Mažas nuostolių skaičius reiškia, kad modelis yra labai tikslus ir geba pateikti tikslas veikimo prognozes, kaip numatyta **18 pav.** Tai ypač svarbu atliekant tikslumo reikalaujančias užduotis, pavyzdžiui, priimant gyvybiškai svarbius sprendimus arba taikant jautrias programas.

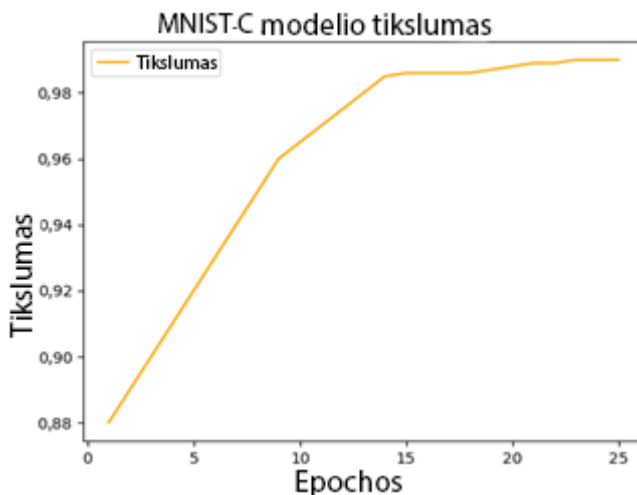


18 pav. Anomalijų aptikimo, naudojant MNIST-C duomenų rinkinį, rezultatų prognozinis grafikas

Modelis buvo apmokytas ir išbandytas per 25 epochas, kad būtų pasiektas didelis tikslumas ir minimalūs nuostoliai, kaip parodyta **19** ir **20 pav.** Modelis gana gerai veikia nuostolių atžvilgiu. Nuostoliai tolygiai mažėja nuo 0,45 pirmoje epochoje iki 0,06 paskutinėje epochoje, kaip matyti iš grafiko „MNIST-C modelio nuostoliai“. Ši mažėjimo tendencija reiškia, kad modelis sėkmingai mažina klaidų skaičių ir tiksliau prognozuoja. Nuostolių konvergencija rodo, kad modelis mokosi ir prisitaiko prie mokymo duomenų, todėl laikui bėgant jo rezultatai gerėja.



19 pav. Nekontroliuojamos partijos nuostoliai naudojant MNIST-C duomenų rinkinį



20 pav. Nekontroliuojamos partijos tikslumas naudojant MNIST-C duomenų rinkinį

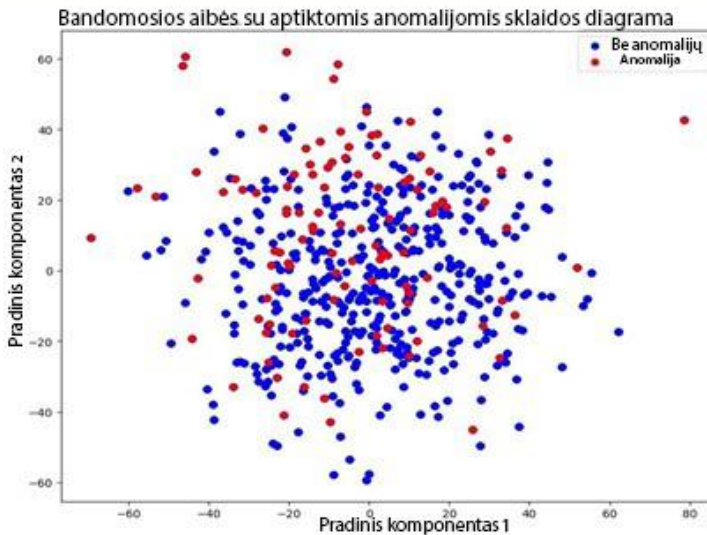
MNIST-C modelio tikslumo grafike rodomas modelio tikslumas mokymo metu, kaip matote **21 pav.** Pirmojoje epochoje jo tikslumas prasideda nuo 0,88 ir su kiekviena tolesne epocha nuolat didėja. Mokymo sesijos pabaigoje tikslumas pasiekia 0,99. Mokymo metu gerėja modelio gebėjimas tiksliai kategorizuoti didesnę duomenų rinkinio nuotraukų dalį. Modelis pasiekia stebėtiną tikslumo lygį, parodydamas savo gebėjimą atskirti įvairius MNIST-C duomenų rinkinio skaitmenis. Bendras našumas: iš nuostolių ir tikslumo rodiklių matyti, kad MNIST-C modelis veikia neįtikėtinai gerai. Modelio gebėjimą šalinti klaidas ir kurti tikslas prognozes rodo mažėjantys nuostolių skaičiai. Be to, modelio gebėjimą veiksmingai kategorizuoti MNIST-C duomenų rinkinio vaizdus rodo nuolat didėjančios tikslumo reikšmės. Modelio nuostolių konvergencija ir didelis tikslumas reiškia, kad jis sėkmingai išmoko duomenų rinkinio schemas ir požymius, todėl gali kurti teisingas prognozes.

11 lentelė. MNIST-C duomenų rinkinio prognozavimo modelio reikšmės

Tikslumas	0,989
Tikrųjų teigiamų rezultatų tikslumas	0,91
Atšaukimas	0,8889
F1 rezultatas	0,8233

Prognozavimo tikslumas yra 0,989, todėl modelis pasižymi aukštu teisingumo lygiu, teisingai nustatydamas 98,9 % atvejų. 0,91 tikslumo įvertis rodo, kad 91 %

prognozuotų teigiamųjų atvejų buvo tikrai teigiamieji, kaip parodyta **11 lentelėje**. Tai žymi modelio gebėjimą sumažinti klaidingų teigiamųjų atvejų skaičių. Atšaukimo, arba jautrumo, įvertis 0,8889 reiškia, kad modelis teisingai nustatė 88,89 % teigiamųjų atvejų iš visų teigiamųjų atvejų. Tai rodo, kaip gerai modelis fiksuoja reikšmingus atvejus. F1 balas, kuris yra tikslumo ir atšaukimo harmoninis vidurkis, žymi bendrą modelio našumą. Modelis pasiekia tinkamą tikslumo ir atšaukimo derinį – F1 rezultatas yra 0,8233. Taigi daroma išvada, kad jis prognozuoja labai tiksliai ir veiksmingai suderina klaidingų teigiamųjų rezultatų mažinimą ir reikšmingų įvykių fiksavimą.



21 pav. MNIST-C modelio tikslumo grafikas

4.4.4 K-MNIST duomenų rinkinys

Pradinėje epochoje modelio nuostoliai yra gana dideli – 0,2506, o patvirtinimo nuostoliai – 0,1835, kaip parodyta **12 lentelėje**. Tačiau mokymui progresuojant tiek nuostoliai, tiek patvirtinimo nuostoliai nuosekliai mažėja. Ši tendencija rodo, kad modelis veiksmingai mokosi ir laikui bėgant gerina savo veikimą. Iki dešimtosios epochos nuostoliai dar labiau sumažėjo iki 0,1488, o patvirtinimo nuostoliai yra 0,1490. Tai rodo, kad modelis pasiekė mažesnę nuostolių ir patvirtinimo nuostolių lygį, vadinasi, jo tikslumas ir apibendrinimas geresni. Mažėjanti patvirtinimo nuostolių vertė, kuri išlieka artima nuostolių vertei, rodo, kad modelis sėkmingai apibendrina išmokus požymius nematytiems duomenims. Apskritai, remiantis mažėjančiais nuostoliais ir patvirtinimo nuostoliais, modelis demonstruoja geresnį veikimą ir gebėjimą užfiksuoti pagrindinius dėsningumus nurodytuose duomenyse.

12 lentelė. K-MNIST duomenų rinkinio nuostolių vertės

Epocha	Nuostoliai	Tikrieji teigiamieji	Tikrieji neigiamieji	Klaidin-gai teigiamieji	Klai-dingai neigiamieji
0	0,45	76 432	108 953	19 583	14 321
1	0,4	80 214	113 482	14 763	10 025
2	0,36	85 297	120 197	10 134	10 073
3	0,32	91 345	123 690	4875	5319
4	0,29	93 987	132 504	1276	3822

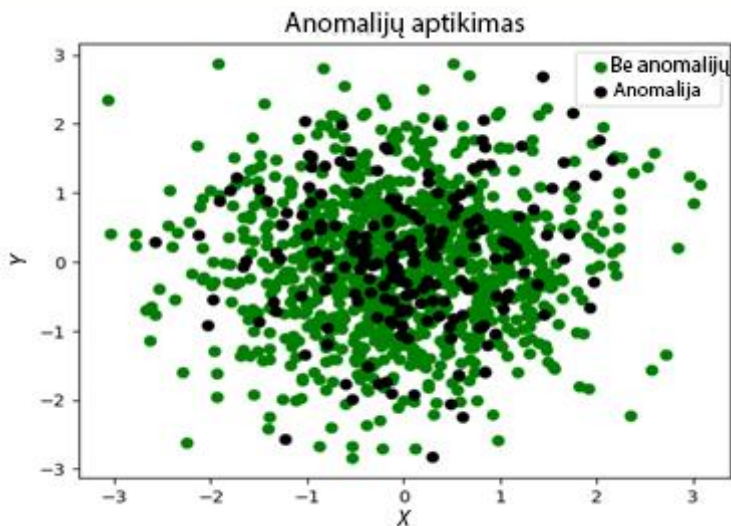
Nuostolių vertė, rodanti prognozuojamų ir faktinių verčių neatitikimą, yra palyginti maža – 0,1488, o tai rodo gerą atitikimą su pagrindiniais teisingais duomenimis. Tikslumas 0,9354 rodo, kad modelis teisingai klasifikuoja maždaug 93,54 % atvejų. Tikslumas, kuris parodo gebėjimą išvengti klaidingai teigiamų rezultatų, yra didelis – 0,9217. Atšaukimas (0,9481) rodo, kad modelis sėkmingai nustatė didelę dalį teigiamųjų atvejų. **13 lentelėje** pateikiama pagrindinių modelio veiklos rodiklių santrauka.

13 lentelė. K-MNIST duomenų rinkinio prognozuojamos vertinimo metrikos

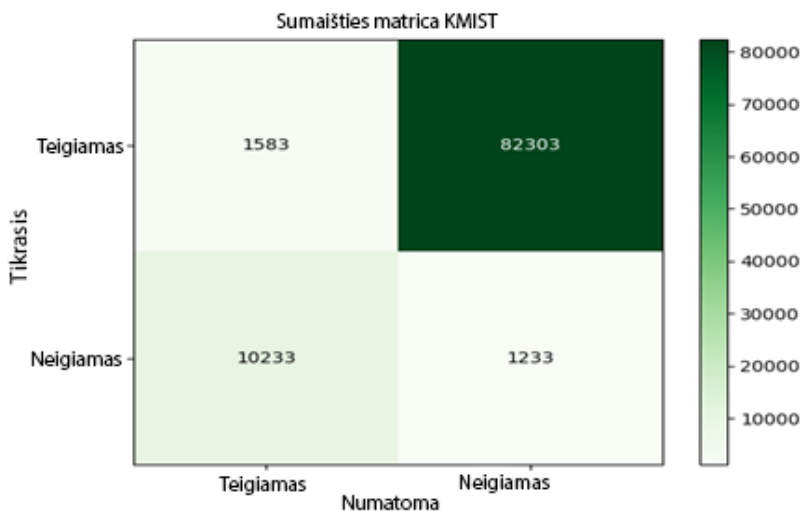
Metrika	Vertė
Nuostoliai	0,1488
Tikslumas	0,9354
Rodymo tikslumas	0,9217
Atšaukimas	0,9481
AUC	0,9772
F1 balas	0,9347

Nuostolių vertė, rodanti prognozuojamų ir faktinių verčių neatitikimą, yra palyginti maža – 0,1488, o tai rodo gerą atitikimą su pagrindiniais teisingais

duomenimis. Tikslumas 0,9354 rodo, kad modelis teisingai suklasifikavo apie 93,54 % atvejų. Tikslumas, kuris parodo gebėjimą išvengti klaidingai teigiamų rezultatų, yra didelis – 0,9217. Atšaukimas (0,9481) rodo, kad modelis sėkmingai nustatė didelę dalį teigiamųjų atvejų. AUC (ploto po kreive) reikšmė 0,9772 atspindi gerą modelio skiriamąjį gebėjimą, o F1 balas 0,9347 rodo subalansuotą tikslumo ir atšaukimo efektyvumą. Apskritai modelis pasižymi perspektyviu veikimu pagal kelis vertinimo rodiklius. **22 pav.** pavaizduoti modelio tikslumo rezultatai grafiko formatu.

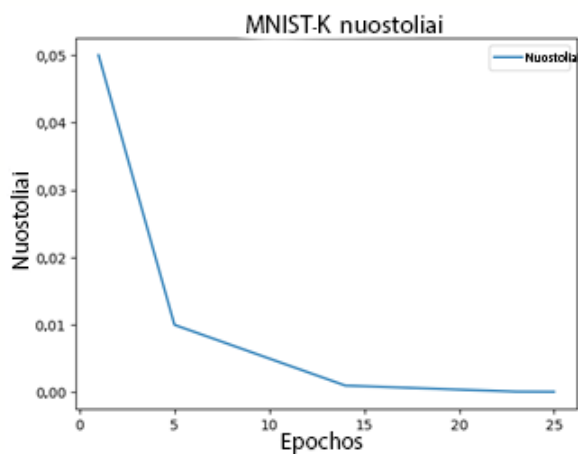


22 pav. K-MNIST modelio tikslumo grafikas

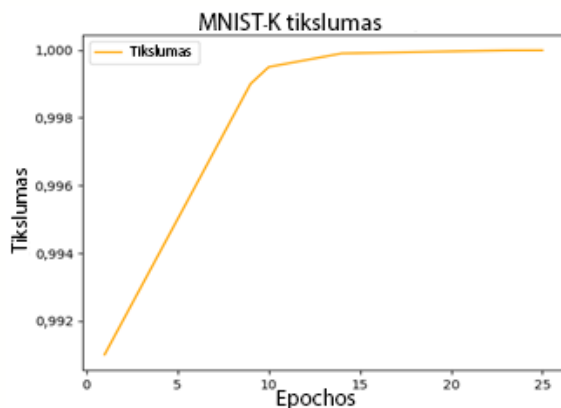


23 pav. Prognozuojamas tikslumo rezultatas K-MNIST duomenų rinkinio sumaišties matrica

23 pav. parodytas K-MNIST duomenų rinkiniui taikomos prognozavimo sumaišties matricos tikslumo rezultatas, kurio faktinės teigiamosios ir neigiamosios reikšmės buvo atitinkamai 1583 ir 10 233. Pagal prognozuojamą vertę rezultatai yra 82 303 teigiamieji ir 1233 neigiamieji. **24 ir 25 pav.** pavaizduotas modelio mokymas ir testavimas per 25 epochas, siekiant didelio tikslumo ir minimalių nuostolių. K-MNIST duomenų rinkinio **26 ir 27 pav.** parodyti atitinkamai tipinių vaizdų ir nustatytų anomalijų pavyzdžiai.



24 pav. Nekontroliuojamos partijos nuostoliai naudojant MNIST-K duomenų rinkinį



25 pav. Nekontroliuojamos partijos tikslumas naudojant MNIST-K duomenų rinkinį



26 pav. K-MNIST duomenų rinkinio vaizdų pavyzdžiai



27 pav. Anomalijos, aptiktos naudojant K-MNIST duomenų rinkinį

4.5 Apibendrinimas

Duomenų saugumui, slaptumui ir vientisumui užtikrinti taikėme pažangiausius metodus, įskaitant skaičiavimus su nulinių žinių įrodymu (ZKP). Norint aptikti duomenų anomalijas ir nuokrypius, buvo taikomi nauji metodai, tokie kaip autokoderis ir atraminių vektorių klasifikatorius (SVM). Išvados buvo išsaugotos įvairiais formatais, įskaitant paveikslėlius, CSV failus bei grafikus, ir suteikė reikšmingų įžvalgų apie nustatytas anomalijas. Darbo įrodymo (PoW) konsensuso mechanizmas patobulino blokų grandinės tinklus, užtikrindamas saugias ir klastojimui atsparias sandorių istorijas. Mūsų modeliams apmokyti buvo taikomi ir prižiūravimo, ir neprižiūravimo mokymosi metodai. Remiantis įrodytais įvesties ir išvesties ryšiais, prižiūrimasis mokymasis naudojant paženklintus duomenis leido atlikti patikimas prognozes. Neprižiūrimasis mokymasis atskleidė anksčiau nežinomus modelius ir struktūras neženklinčiuose duomenyse. Integruodami šias metodikas sukūrėme pritaikomus ir patikimus mašininio mokymosi modelius. Šis pažangių metodų, tokių kaip ZKP skaičiavimai, geresni anomalijų aptikimo algoritmai, PoW konsensusas ir hibridinė mokymosi metodika, derinys leido sukurti saugią ir veiksmingą duomenų analizės sistemą, kuri nepažeidžia privatumo ir duomenų vientisumo. Išsamus kompiuterinės regos etalono MNIST-C duomenų rinkinio testavimas patvirtino puikų mūsų modelio veikimą. Jo tikslumas siekė 0,989, ir jis teisingai klasifikavo daugumą įvykių. Tikslumo rodiklis 0,91 rodo, kad modelis teisingai nuspėjo didelę dalį teigiamųjų pavyzdžių, o atšaukimo reikšmė 0,8889 rodo modelio gebėjimą atpažinti nemažai teigiamųjų įvykių. Subalansuotas F1 balas 0,8233 dar kartą įrodo mūsų metodo veiksmingumą anomalijų aptikimo darbams. Apskritai nuoseklus mūsų modelio veikimas ir sėkmingas mūsų metodikos

taikymas rodo jos naudingumą realiomis sąlygomis. Derindami pažangiausius metodus sukūrėme pagrindą patikimai ir saugiai duomenų analizei įvairiose srityse. Puikūs veiklos rodikliai patvirtina mūsų metodo aktualumą ir pritaikomumą, atverdami kelią tolesniems tyrimams ir praktiniam taikymui.

5. Išvados

1. Esamų mokslinių publikacijų analizė parodė, kad šiuo metu taikomi anomalijų aptikimo metodai susiduria su sunkumais. Šie sunkumai apima anomalijų nustatymą, didžiulių duomenų kiekių valdymą, aiškaus modelio sukūrimą, sudėtingų analitinių galimybių suteikimą, duomenų tikslumo užtikrinimą ir proceso lankstumo užtikrinimą. Literatūros tyrimai taip pat parodė, kad turimos anomalijų aptikimo ontologijos kelios šalys gali taikyti federacinį mokymąsi, kad kolektyviai apmokytų visuotinį modelį pagal savo vietinius duomenų rinkinius, aiškiai nesikeisdamos duomenimis. Kadangi svarbiausia informacija lieka vietiniuose įrenginiuose, ši strategija užtikrina duomenų privatumą ir saugumą.
2. Atsižvelgiant į literatūros tyrimų analizės rezultatus ir į literatūros analizės išvadas, siūlomas unikalūs anomalijų atpažinimo vaizdų duomenų rinkiniuose metodas, taikant federacinį mokymąsi ir blokų grandinės technologiją. Siekiant dar labiau padidinti privatumą ir saugumą, įtraukti nulinių žinių įrodymo (ZKP) metodai, leidžiantys šalims išbandyti savo vietinių modelių tikslumą nepažeidžiant savo privačių duomenų. Siūloma metodika, derinant prižiūrimojo ir neprižiūrimojo mokymosi metodus, identifikuoja ir klasifikuoja anomalijas vaizdų duomenų rinkiniuose didžiausiu tikslumu.
3. Siūlomu metodu sėkmingai atsižvelgiama į visus anomalijų aptikimui vaizduose svarbius veiksnius ir požymius, todėl atliekama išsami anomalijų modelių analizė. Sėkmingai gauta ir pateikta informacija apie vaizdų duomenų rinkinio problemas, tokias kaip nukrypimai nuo numatytų modelių, netikėti vaizdo bruožai ir vizualiųjų anomalijų nustatymas. Palyginti su ankstesniais metodais, pasiūlytas metodas suteikė išsamesnių žinių apie anomalijas, nes apėmė daugybę savybių ir požymių, būdingų vaizdų anomalijų identifikavimui.
4. Metodas buvo sukurtas *Python 3.11* versijos aplinkoje, kad būtų užtikrintas geriausias mastelio keitimas ir skaičiavimo efektyvumas. Siekiant užtikrinti dinaminį našumą, kodas buvo vykdomas naudojant *NVIDIA T4 Tensor Core GPU* skaičiavimo pajėgumus, leidžiančius greitai ir tiksliai apdoroti anomalijų aptikimo algoritmus. Čia pasiūlytas unikalūs anomalijų aptikimo sprendimas turi gerai struktūrizuotą architektūrą, pagrįstą UML projektavimo principais.
5. Anomalijų aptikimas pagerėjo, išmokius mašininio mokymosi modelį pagal kelių tinklo mazgų duomenis. Tai leido modeliui išmokyti įprastas tinklo veiklos schemas ir tiksliai aptikti bet kokius nukrypimus nuo šių schemų, kurie rodo anomalius duomenis. Dėl šios priežasties buvo sumažintos skaičiavimo sąnaudos taikant federacinį mokymą, kuris leidžia modelį mokyti pačiuose mazguose, o ne perduoti visus duomenis į centrinę mokymo vietą. Pasiūlytas metodas užtikrina geresnį mastelio keitimą, nes duomenys renkami iš kelių mazgų, todėl modeliui apmokyti galima naudoti daugiau duomenų pavyzdžių.

LITERATURE

1. BISWAS, K., & MISRA, S. (2019). Blockchain for supply chain traceability: A systematic literature review. *Computers & Industrial Engineering*, 137, 106070.2.
2. DORRI, A., KANHERE, S. S., JURDAK, R., & GAURAVARAM, P. (2017). Blockchain for IoT security and privacy: The case study of a smart home. In 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 618-623). IEEE.
3. IANSITI, MARCO, AND KARIM R. LAKHANI. "The truth about blockchain." *Harvard business review* 95, no. 1 (2017): 118-127.
4. MOUGAYAR, WILLIAM. *The business blockchain: promise, practice, and application of the next Internet technology*. John Wiley & Sons, 2016.
5. NAKAMOTO, SATOSHI. "Bitcoin: A peer-to-peer electronic cash system." *Decentralized business review* (2008): 21260.
6. SWAN, MELANIE. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.
7. TAPSCOTT, DON, AND ALEX TAPSCOTT. *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Penguin, 2016.
8. BHUTTA, M. N. M., KHWAJA, A. A., NADEEM, A., AHMAD, H. F., KHAN, M. K., HANIF, M. A., & CAO, Y. (2021). A survey on blockchain technology: evolution, architecture, and security. *IEEE Access*, 9, 61048-61073.
9. SAI, A. R., BUCKLEY, J., FITZGERALD, B., & LE GEAR, A. (2021). Taxonomy of centralization in public blockchain systems: A systematic literature review. *Information Processing & Management*, 58(4), 102584.
10. POP, C., CIOARA, T., ANGHEL, I., ANTAL, M., & SALOMIE, I. (2020). Blockchain based decentralized applications: Technology review and development guidelines. *arXiv preprint arXiv:2003.07131*.
11. POP, CLAUDIA DANIELA, MARCEL ANTAL, TUDOR CIOARA, IONUT ANGHEL, AND IOAN SALOMIE. "Blockchain and demand response: Zero-knowledge proofs for energy transactions privacy." *Sensors* 20, no. 19 (2020): 5678.
12. HSIEH, Y. Y., VERGNE, J. P., ANDERSON, P., LAKHANI, K., & REITZIG, M. (2018). Bitcoin and the rise of decentralized autonomous organizations. *Journal of Organization Design*, 7(1), 1-16.
13. ZHAI, SHEPING, YUANYUAN YANG, JING LI, CHENG QIU, AND JIANGMING ZHAO. "Research on the Application of Cryptography on the Blockchain." In *Journal of Physics: Conference Series*, vol. 1168, no. 3, p. 032077. IOP Publishing, 2019.
14. ZHENG, Z., XIE, S., DAI, H. N., CHEN, X., & WANG, H. (2018). Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4), 352-375.
15. WU, K., MA, Y., HUANG, G., & LIU, X. (2021). A first look at blockchain-based decentralized applications. *Software: Practice and Experience*, 51(10), 2033-2050.
16. RAJA SANTHI, A., & MUTHUSWAMY, P. (2022). Influence of blockchain technology in manufacturing supply chain and logistics. *Logistics*, 6(1), 15.
17. MERKLE, R. C. (1987, August). A digital signature based on a conventional encryption function. In Conference on the theory and application of cryptographic techniques (pp. 369-378). Springer, Berlin, Heidelberg.

18. ABRAHAM, ITTAI, AND DAHLIA MALKHI. "The blockchain consensus layer and BFT." *Bulletin of EATCS* 3, no. 123 (2017)].
19. MAHMOOD, Z., & VACIUS, J. (2020, December). Privacy-Preserving Block-chain Framework Based on Ring Signatures (RSs) and Zero-Knowledge Proofs (ZKPs). In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)* (pp. 1-6). IEEE.
20. WANG, PING, AND SHUHAN QIAO. "Emerging applications of blockchain technology on a virtual platform for English teaching and learning." *Wireless Communications and Mobile Computing* 2020.
21. RAY, P. P., DASH, D., SALAH, K., & KUMAR, N. (2020). Blockchain for IoT-based healthcare: background, consensus, platforms, and use cases. *IEEE Systems Journal*, 15(1), 85-94.
22. MIERS, IAN, CHRISTINA GARMAN, MATTHEW GREEN, AND AVIEL D. RUBIN. "ZeroCoin: Anonymous distributed e-cash from bitcoin." In *2013 IEEE Symposium on Security and Privacy*, pp. 397-411. IEEE, 2013.
23. MORRIS, VANCE, ROHIT ADIVI, RATNAKAR ASARA, MATTHEW COUSENS, NICK GUPTA, NICHOLAS LINCOLN, BARRY MOSAKOWSKI, AND HONG WEI SUN. *Developing a Blockchain Business Network with Hyperledger Composer using the IBM Blockchain Platform Starter Plan*. IBM Redbooks, 2018.
24. ANILKUMAR, V., JOJI, J. A., AFZAL, A., & SHEIK, R. (2019, May). Blockchain simulation and development platforms: survey, issues and challenges. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (pp. 935-939). IEEE.
25. BAMBARA, JOSEPH J., AND PAUL R. ALLEN. "Blockchain." *A practical guide to developing business, law and technology solutions*. New York City: McGraw-Hill Professional (2018).
26. MARTINS, JOSE P., JOAO C. FERREIRA, VITOR MONTEIRO, JOSE A. AFONSO, AND JOAO L. AFONSO. "IoT and blockchain paradigms for EV charging systems." *Energies* 12, no. 15 (2019): 2987.
27. WANG, JIN, BOYANG WEI, JINGYU ZHANG, XIAOFENG YU, AND PRADIP KUMAR SHARMA. "An optimized transaction verification method for trustworthy blockchain-enabled IIoT." *Ad Hoc Networks* 119 (2021): 102526.
28. OLIVA, GUSTAVO A., AHMED E. HASSAN, AND ZHEN MING JIANG. "An exploratory study of smart contracts in the Ethereum blockchain platform." *Empirical Software Engineering* 25 (2020): 1864-1904.
29. KANIMOZHI, E., AND D. AKILA. "Block chain smart contracts on Iot." *International Journal of Recent Technology and Engineering* 8, no. 2 (2019): 105-110.
30. MAHMOOD, Z., & VACIUS, J. (2020, December). Privacy-Preserving Block-chain Framework Based on Ring Signatures (RSs) and Zero-Knowledge Proofs (ZKPs). In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)* (pp. 1-6). IEEE.
31. D. BONEH et al., "Secure Password-less Sign-In Using Continuous Authentication," Proceedings of the 2016 ACM SIGSAC Conference on Computer & Communications Security, 2016.
32. LOKHAVA, MARTA, GIULIANO LOSA, DAVID MAZIÈRES, GRAYDON HOARE, NICOLAS BARRY, ELI GAFNI, JONATHAN JOVE, RAFAŁ MALINOWSKY, AND JED MCCAULEB. "Fast and secure global payments with

- stellar." In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, pp. 80-96. 2019.
33. JUELS, "Zero-Knowledge Proofs of Identity," *Proceedings of the 2014 ACM SIGSAC Conference on Computer & Communications Security*, 2014.
 34. CERECEDO, M., MATSUMOTO, T., & IMAI, H. (1993). Efficient and secure multiparty generation of digital signatures based on discrete logarithms. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 76(4), 532-545.
 35. CHAUM, DAVID. "Security without identification: Transaction systems to make big brother obsolete." *Communications of the ACM* 28, no. 10 (1985): 1030-1044.
 36. LASHKARI, B., & MUSILEK, P. (2021). A comprehensive review of blockchain consensus mechanisms. *IEEE Access*, 9, 43620-43652.
 37. KIAYIAS, AGGELOS, AND DIONYSIS ZINDROS. "Proof-of-work sidechains." In *International Conference on Financial Cryptography and Data Security*, pp. 21-34. Springer, Cham, 2019.
 38. LI, WENTING, SÉBASTIEN ANDREINA, JENS-MATTHIAS BOHLI, AND GHASSAN KARAME. "Securing proof-of-stake blockchain protocols." In *Data privacy management, cryptocurrencies and blockchain technology*, pp. 297-315. Springer, Cham, 2017.
 39. SNIDER, M., SAMANI, K., & JAIN, T. (2018). Delegated proof of stake: features & tradeoffs. *Multicoins Cap*, 19.
 40. RAMKUMAR, N., G. SUDHASADASIVAM, AND K. G. SARANYA. "A survey on different consensus mechanisms for the blockchain technology." In *2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0458-0464. IEEE, 2020.
 41. LIANG, X., ZHAO, J., SHETTY, S., LIU, J., & LI, D. (2017, October). Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)* (pp. 1-5). IEEE.
 42. XU, JIE, BENJAMIN S. GLICKSBERG, CHANG SU, PETER WALKER, JIANG BIAN, AND FEI WANG. "Federated learning for healthcare informatics." *Journal of Healthcare Informatics Research* 5 (2021): 1-19.
 43. FLAVIÁN, C., & GUINALÍU, M. (2006). Consumer trust, perceived security and privacy policy: three basic elements of loyalty to a web site. *Industrial management & data Systems*.
 44. POHLE, J. (2018). Data Privacy Legislation In The European Union Member States—A Practical Overview. *Computer Law Review International*, 19(4), 97-116.
 45. ZHANG, CHEN, YU XIE, HANG BAI, BIN YU, WEIHONG LI, AND YUAN GAO. "A survey on federated learning." *Knowledge-Based Systems* 216 (2021): 106775.
 46. JAYABALAN, JAYAPRIYA, AND N. JEYANTHI. "A study on distributed consensus protocols and algorithms: the backbone of blockchain networks." In *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-10. IEEE, 2021.
 47. NIRANJANAMURTHY, M., B. N. NITHYA, AND S. J. C. C. JAGANNATHA. "Analysis of Blockchain technology: pros, cons and SWOT." *Cluster Computing* 22 (2019): 14743-14757.
 48. QAMMAR, A., KARIM, A., NING, H., & DING, J. (2022). Securing federated learning with blockchain: a systematic literature review. *Artificial Intelligence Review*, 1-35.

49. ZHANG, WEISHAN, QINGHUA LU, QIUYU YU, ZHAOTONG LI, YUE LIU, SIN KIT LO, SHIPING CHEN, XIWEI XU, AND LIMING ZHU. "Blockchain-based federated learning for device failure detection in industrial IoT." *IEEE Internet of Things Journal* 8, no. 7 (2020): 5926-5937.
50. QU, Y., GAO, L., LUAN, T. H., XIANG, Y., YU, S., LI, B., & ZHENG, G. (2020). Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet of Things Journal*, 7(6), 5171-5183.
51. LU, YUNLONG, XIAOHONG HUANG, YUEYUE DAI, SABITA MAHARJAN, AND YAN ZHANG. "Federated learning for data privacy preservation in vehicular cyber-physical systems." *IEEE Network* 34, no. 3 (2020): 50-56.
52. BELOTTI, M., BOŽIĆ, N., PUJOLLE, G., & SECCI, S. (2019). A vademecum on blockchain technologies: When, which, and how. *IEEE Communications Surveys & Tutorials*, 21(4), 3796-3838.
53. ZHOU, ZHI, XU CHEN, EN LI, LIEKANG ZENG, KE LUO, AND JUNSHAN ZHANG. "Edge intelligence: Paving the last mile of artificial intelligence with edge computing." *Proceedings of the IEEE* 107, no. 8 (2019): 1738-1762.
54. JAVED, ABDUL REHMAN, MUHAMMAD ABUL HASSAN, FAISAL SHAHZAD, WAQAS AHMED, SAURABH SINGH, THAR BAKER, AND THIPPA REDDY GADEKALLU. "Integration of blockchain technology and federated learning in vehicular (iot) networks: A comprehensive survey." *Sensors* 22, no. 12 (2022): 4394.
55. MAHMOOD, Z., & JUSAS, V. (2022). Blockchain-Enabled: Multi-Layered Security Federated Learning Platform for Preserving Data Privacy. *Electronics*, 11(10), 1624.
56. LANE, TERRAN D. *Machine learning techniques for the computer security domain of anomaly detection*. Purdue University, 2000.
57. ZIMEK, ARTHUR, AND PETER FILZMOSER. "There and back again: Outlier detection between statistical reasoning and data mining algorithms." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, no. 6 (2018): e1280.
58. RASHID, AQSA, ASIF MASOOD, HAIDER ABBAS, AND YIN ZHANG. "Blockchain-based public key infrastructure: a transparent digital certification mechanism for secure communication." *IEEE Network* 35, no. 5 (2021): 220-225.
59. AHMED, MOHIUDDIN, ABDUN NASER MAHMOOD, AND JIANKUN HU. "A survey of network anomaly detection techniques." *Journal of Network and Computer Applications* 60 (2016): 19-31.
60. ESKIN, E., ARNOLD, A., PRERAU, M., PORTNOY, L., & STOLFO, S. (2002). A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Applications of data mining in computer security*, 77-101.
61. OGUNDOKUN, R. O., MISRA, S., MASKELIUNAS, R., & DAMASEVICIUS, R. (2022). A Review on Federated Learning and Machine Learning Approaches: Categorization, Application Areas, and Blockchain Technology. *Information*, 13(5), 263.
62. OYMAK, SAMET, AND TALHA CIHAD GULCU. "A theoretical characterization of semi-supervised learning with self-training for gaussian mixture models." In *International Conference on Artificial Intelligence and Statistics*, pp. 3601-3609. PMLR, 2021.
63. BENJI, M., & SINDHU, M. (2019). A study on the Corda and Ripple blockchain platforms. In *Advances in Big Data and Cloud Computing: Proceedings of ICBDC18* (pp. 179-187). Springer Singapore.

64. MEYER, D., & WIEN, F. T. (2015). Support vector machines. The Interface to libsvm in package e1071, 28, 20.
65. QIU, JUNFEI, QIHUI WU, GUORU DING, YUHUA XU, AND SHUO FENG. "A survey of machine learning for big data processing." *EURASIP Journal on Advances in Signal Processing* 2016 (2016): 1-16.
66. KLASSEN, MYUNGSOOK, NYUNSU KIM, AND WEI MING LIU. "Empirical Study of Support Vector Machine Kernels with Applications to Microarray Data." In *CATA*, pp. 25-30. 2010.
67. ALLAHYARI, MEHDI, SEYEDAMIN POURIYEH, MEHDI ASSEFI, SAIED SAFAEI, ELIZABETH D. TRIPPE, JUAN B. GUTIERREZ, AND KRYS KOCHUT. "A brief survey of text mining: Classification, clustering and extraction techniques." *arXiv preprint arXiv:1707.02919* (2017).
68. LI, JINBO, HESAM IZAKIAN, WITOLD PEDRYCZ, AND IQBAL JAMAL. "Clustering-based anomaly detection in multivariate time series data." *Applied Soft Computing* 100 (2021): 106919.
69. RAMCHANDRAN, A., & SANGAIAH, A. K. (2018). Unsupervised anomaly detection for high dimensional data—An exploratory analysis. In *Computational intelligence for multimedia big data on the cloud with engineering applications* (pp. 233-251). Academic Press.
70. HUANG, ZHONGLING, ZONGXU PAN, AND BIN LEI. "Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data." *Remote Sensing* 9, no. 9 (2017): 907.
71. LIU, HAIZHOU, XUAN ZHANG, XINWEI SHEN, AND HONGBIN SUN. "A federated learning framework for smart grids: Securing power traces in collaborative learning." *arXiv preprint arXiv:2103.11870* (2021).
72. WEI, YUANYUAN, JULIAN JANG-JACCARD, FARIZA SABRINA, WEN XU, SEYIT CAMTEPE, AND AERYN DUNMORE. "Reconstruction-based LSTM-Autoencoder for Anomaly-based DDoS Attack Detection over Multivariate Time-Series Data." *arXiv preprint arXiv:2305.09475* (2023).
73. DEY, SAMRAT KUMAR, AND MD MAHBUBUR RAHMAN. "Effects of machine learning approach in flow-based anomaly detection on software-defined networking." *Symmetry* 12, no. 1 (2019): 7.
74. KARMOUS, N., AOUEILEYINE, M. O. E., ABDELKADER, M., & YOUSSEF, N. (2022, March). A Proposed Intrusion Detection Method Based on Machine Learning Used for Internet of Things Systems. In *Advanced Information Networking and Applications: Proceedings of the 36th International Conference on Advanced Information Networking and Applications (AINA-2022), Volume 3* (pp. 33-45). Cham: Springer International Publishing.
75. FENG, CHAOSHENG, BIN LIU, KEPING YU, SOTIRIOS K. GOUDOS, AND SHAOHUA WAn. "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs." *IEEE Transactions on Industrial Informatics* 18, no. 5 (2021): 3582-3592.
76. SATER, RAED ABDEL, AND A. BEN HAMZA. "A federated learning approach to anomaly detection in smart buildings." *ACM Transactions on Internet of Things* 2, no. 4 (2021): 1-23.
77. LU, C., WANG, Z. Y., QIN, W. L., & MA, J. (2017). Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Processing*, 130, 377-388.

78. YEGANEH, Y., FARSHAD, A., NAVAB, N., & ALBARQOUNI, S. (2020). Inverse distance aggregation for federated learning with non-iid data. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 2* (pp. 150-159). Springer International Publishing.
79. DE SELLER, MICAH J., BRANDON EDWARDS, G. ANTHONY REINA, JASON MARTIN, SARTHAK PATI, AIKATERINI KOTROTSOU, MIKHAIL MILCHENKO et al. "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data." *Scientific reports* 10, no. 1 (2020): 1-12.
80. GHIMIRE, BIMAL, AND DANDA B. RAWAT. "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things." *IEEE Internet of Things Journal* (2022).
81. BOUACIDA, NADER, AND PRASANT MOHAPATRA. "Vulnerabilities in federated learning." *IEEE Access* 9 (2021): 63229-63249.
82. HASSAN, MUNEEB UL, MUBASHIR HUSAIN REHMANI, AND JINJUN CHEN. "Anomaly detection in blockchain networks: A comprehensive survey." *IEEE Communications Surveys & Tutorials* (2022).
83. LV, P., XIE, L., XU, J., WU, X., & LI, T. (2022). Misbehavior Detection in Vehicular Ad Hoc Networks Based on Privacy-Preserving Federated Learning and Blockchain. *IEEE Transactions on Network and Service Management*.
84. KESHK, MARWA, BENJAMIN TURNBULL, NOUR MOUSTAFA, DINUSHA VATSALAN, AND KIM-KWANG RAYMOND CHOO. "A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks." *IEEE Transactions on Industrial Informatics* 16, no. 8 (2019): 5110-5118.
85. MAHMOOD, ZEB, AND VACIUS JUSAS. "Implementation framework for a blockchain-based federated learning model for classification problems." *Symmetry* 13, no. 7 (2021): 1116.
86. SELIM, GAMAL ELDIN I., EZZ EL-DIN HEMDAN, AHMED M. SHEHATA, AND NAWAL A. EL-FISHAWY. "Anomaly events classification and detection system in critical industrial internet of things infrastructure using machine learning algorithms." *Multimedia Tools and Applications* 80 (2021): 12619-12640.
87. WANG, XIAODING, JIA HU, HUI LIN, WENXIN LIU, HYEONJOON MOON, AND MD JALIL PIRAN. "Federated Learning-Empowered Disease Diagnosis Mechanism in the Internet of Medical Things: From the Privacy-Preservation Perspective." *IEEE Transactions on Industrial Informatics* (2022).
88. KAMIŠALIĆ, A., KRAMBERGER, R., & FISTER JR, I. (2021). Synergy of blockchain technology and data mining techniques for anomaly detection. *Applied Sciences*, 11(17), 7987.
89. WEN, MI, RONG XIE, KEJIE LU, LIANGLIANG WANG, AND KAI ZHANG. "FedDetect: a novel privacy-preserving federated learning framework for energy theft detection in smart grid." *IEEE Internet of Things Journal* 9, no. 8 (2021): 6069-6080.
90. LI, BEIBEI, SHANG MA, RUILONG DENG, KIM-KWANG RAYMOND CHOO, AND JIN YANG. "Federated anomaly detection on system logs for the internet of things: A customizable and communication-efficient approach." *IEEE Transactions on Network and Service Management* 19, no. 2 (2022): 1705-1716.

91. FAHIM, MUHAMMAD, AND ALBERTO SILLITTI. "Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review." *IEEE Access* 7 (2019): 81664-81681.
92. CUI, LEI, YOUYANG QU, GANG XIE, DEZE ZENG, RUIDONG LI, SHIGEN SHEN, AND SHUI YU. "Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures." *IEEE Transactions on Industrial Informatics* 18, no. 5 (2021): 3492-3500.
93. GUPTA, DEEPTI, OLUMIDE KAYODE, SMRITI BHATT, MAANAK GUPTA, and ALI SAMAN TOSUN. "Hierarchical federated learning based anomaly detection using digital twins for smart healthcare." In *2021 IEEE 7th International Conference on Collaboration and Internet Computing (CIC)*, pp. 16-25. IEEE, 2021.
94. KUMAR, KP SANAL, S. ANU H. NAIR, DEEPSUBHRA GUHA ROY, B. RAJALINGAM, and R. SANTHOSH KUMAR. "Security and privacy-aware artificial intrusion detection system using federated machine learning." *Computers & Electrical Engineering* 96 (2021): 107440.
95. NUR, A. Y. (2021, April). Combating DDoS attacks with fair rate throttling. In *2021 IEEE International Systems Conference (SysCon)* (pp. 1-8). IEEE.
96. PREUVENEERS, DAVY, VERA RIMMER, ILIAS TSINGENOPOULOS, JAN SPOOREN, WOUTER JOOSEN, and ELISABETH ILIE-ZUDOR. "Chained anomaly detection models for federated learning: An intrusion detection case study." *Applied Sciences* 8, no. 12 (2018): 2663.
97. KESHK, MARWA, BENJAMIN TURNBULL, NOUR MOUSTAFA, DINUSHA VATSALAN, and KIM-KWANG RAYMOND CHOO. "A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks." *IEEE Transactions on Industrial Informatics* 16, no. 8 (2019): 5110-5118.
98. GU, YONGHAO, KAIYUE LI, ZHENYANG GUO, and YONGFEI WANG. "Semi-supervised K-means DDoS detection method using hybrid feature selection algorithm." *IEEE Access* 7 (2019): 64351-64365.
99. CHEN, YANG, JUNZHE ZHANG, and CHAI KIAT YEO. "Privacy-preserving knowledge transfer for intrusion detection with federated deep autoencoding gaussian mixture model." *Information Sciences* 609 (2022): 1204-1220.
100. VINITA, L. JAI, and V. VETRISSELVI. "Federated Learning-based Misbehaviour detection on an emergency message dissemination scenario for the 6G-enabled Internet of Vehicles." *Ad Hoc Networks* 144 (2023): 103153.
101. ZHANG, KAI, YUSHAN JIANG, LEE SEVERSKY, CHENGTAO XU, DAHAI LIU, and HOUBING SONG. "Federated variational learning for anomaly detection in multivariate time series." In *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 1-9. IEEE, 2021.
102. CAVALLIN, FLORENCIA, and RUDOLF MAYER. "Anomaly Detection from Distributed Data Sources via Federated Learning." In *Advanced Information Networking and Applications: Proceedings of the 36th International Conference on Advanced Information Networking and Applications (AINA-2022)*, Volume 2, pp. 317-328. Cham: Springer International Publishing, 2022.
103. MOUDOUD, HAJAR, and SOUMAYA CHERKAoui. "Toward Secure and Private Federated Learning for IoT using Blockchain." In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pp. 4316-4321. IEEE, 2022.

- 104.DIEZ-OLIVAN, ALBERTO, JAVIER DEL SER, DIEGO GALAR, and BASILIO SIERRA. "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0." *Information Fusion* 50 (2019): 92-111.
- 105.SUN, XIN, ZHIJUN TANG, MENGXUAN DU, CHAOPING DENG, WENBIN LIN, JINSHAN CHEN, QI QI, and HAIFENG ZHENG. "A Hierarchical Federated Learning-Based Intrusion Detection System for 5G Smart Grids." *Electronics* 11, no. 16 (2022): 2627.
- 106.ARFAOUI, DIPL-ING GHAITH. "Privacy-Preserving Collaborative Anomaly Detection." PhD diss., Wien, 2023.
- 107.CHEN, NAIYUE, YI JIN, YINGLONG LI, and LUXIN CAI. "Trust-based federated learning for network anomaly detection." In *Web Intelligence*, vol. 19, no. 4, pp. 317-327. IOS Press, 2021.
- 108.MOTHUKURI, VIRAAJI, PRACHI KHARE, REZA M. PARIZI, SEYEDAMIN POURIYEH, ALI DEGHANTANHA, and GAUTAM SRIVASTAVA. "Federated-learning-based anomaly detection for iot security attacks." *IEEE Internet of Things Journal* 9, no. 4 (2021): 2545-2554.
- 109.KUMAR, PRABHAT, GOVIND P. GUPTA, and RAKESH TRIPATHI. "TP2SF: A Trustworthy Privacy-Preserving Secured Framework for sustainable smart cities by leveraging blockchain and machine learning." *Journal of Systems Architecture* 115 (2021): 101954.
- 110.ANTON, SIMON DUQUE, SUNEETHA KANOOR, DANIEL FRAUNHOLZ, and HANS DIETER SCHOTTEN. "Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set." In *Proceedings of the 13th international conference on availability, reliability and security*, pp. 1-9. 2018.
- 111.BOGDANOV, DAN, LIINA KAMM, SVEN LAUR, PILLE PRUULMANN-VENGERFELDT, RIIVO TALVISTE, and JAN WILLEMSON. "Privacy-preserving statistical data analysis on federated databases." In *Privacy Technologies and Policy: Second Annual Privacy Forum, APF 2014, Athens, Greece, May 20-21, 2014. Proceedings* 2, pp. 30-55. Springer International Publishing, 2014.
- 112.FRIHA, OTHMANE, MOHAMED AMINE FERRAG, LEI SHU, LEANDROS MAGLARAS, KIM-KWANG RAYMOND CHOO, AND MEHDI NAFAA. "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things." *Journal of Parallel and Distributed Computing* 165 (2022): 17-31.
- 113.NGUYEN, T. D., MARCHAL, S., MIETTINEN, M., FEREIDOONI, H., ASOKAN, N., & SADEGHI, A. R. (2019, July). D²IoT: A federated self-learning anomaly detection system for IoT. In *2019 IEEE 39th International conference on distributed computing systems (ICDCS)* (pp. 756-767). IEEE.
- 114.CHEN, SHIJIN, WILLY SUSILO, YUDI ZHANG, BO YANG, and MINGWU ZHANG. "Privacy-preserving anomaly counting for time-series data in edge-assisted crowdsensing." *Computer Standards & Interfaces* 85 (2023): 103707.
- 115.LECUN, Y., BOTTOU, L., BENGIO, Y., HAFFNER, P., et al.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), 2278–2324 (1998)
- 116.COHEN, GREGORY, SAEED AFSHAR, JONATHAN TAPSON, and ANDRE VAN SCHAIK. "EMNIST: Extending MNIST to handwritten letters." In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921-2926. IEEE, 2017.
- 117.XIAO, H., RASUL, K., VOLLGRAF, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)

118. MU, NORMAN, and JUSTIN GILMER. "Mnist-c: A robustness benchmark for computer vision." *arXiv preprint arXiv:1906.02337* (2019).
119. CLANUWAT, T., BOBER-IRIZAR, M., KITAMOTO, A., LAMB, A., YAMAMOTO, K., HA, D.: Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718* (2018)

CURRICULUM VITAE

Personal details:

Full Name: Zeba Mahmood

Email: Zeba.mehmood@gmail.com

Education:

2009 – 2014 - Bachelor of Science: Computer Science Superior Group of Colleges - Lahore, Pakistan.

2015 – 2017 - Master of Science: Computer Science Superior Group of Colleges - Lahore, Pakistan.

2018 – 2023 - Doctoral studies in Informatics at Kaunas University of Technology.

LIST OF PUBLICATIONS

1. **Mahmood, Zeba;** Jusas, Vacius. Blockchain-enabled: multi-layered security federated learning platform for preserving data privacy // Electronics. Basel: MDPI. ISSN 2079-9292. 2022, vol. 11, iss. 10, art. no. 1624, p. 1-21. DOI: 10.3390/electronics11101624. [Science Citation Index Expanded (Web of Science); Scopus; Dimensions] [IF: 2,690; AIF: 4,505; IF/AIF: 0,597; Q3 (2021, InCites JCR SCIE)] [FOR: N 009] [Input: 0,500].
2. **Mahmood, Zeba;** Jusas, Vacius. Implementation framework for a blockchain-based federated learning model for classification problems // Symmetry. Basel: MDPI. ISSN 2073-8994. 2021, vol. 13, iss. 7, art. no. 1116, p. 1-15. DOI: 10.3390/sym13071116. [Science Citation Index Expanded (Web of Science); Scopus; DOAJ] [IF: 2,940; AIF: 7,694; IF/AIF: 0,382; Q2 (2021, InCites JCR SCIE)] [FOR: N 009] [Input: 0,500].

Scientific Conferences:

1. **Mahmood, Zeba.** Impact of Blockchain technology in healthcare sector during COVID-19 pandemic // Computer science & information technology: international conference on AI, machine learning and applications (AIMLA 2021), August 28 ~ 29, 2021, Dubai, UAE / D.C. Wyld, D. Nagamalai (eds). Chennai, Tamil Nadu: AIRCCpublishing corporation, 2021. ISBN 9781925953473. ISSN 2231-5403. 2021, vol. 11, no. 13, p. 75-88. DOI: 10.5121/csit.2021.111306. [FOR: N 009] [Input: 1,000].
2. **Mahmood, Zeba;** Jusas, Vacius. Implementation framework for a blockchain-based reputation and trust system // ICAETA - 2021: international conference on advanced engineering, technology, and applications (ICAETA -2021), July 09-11, 2021, Istanbul, Turkey: book of proceedings / editorial board: A. Jamil, A.A. Hameed. Istanbul: Istanbul Aydin University. ISSN 2752-8340. 2021, p. 55-61. [FOR: N 009] [Input: 0,500].

3. **Mahmood, Zeba;** Jusas, Vacius. Privacy-preserving block-chain framework based on ring signatures (RSs) and zero-knowledge proofs (ZKPs) // International conference on innovation and intelligence for informatics, computing, and technologies (3ICT), 20-21 December 2020, University of Bahrain, Bahrain. Piscataway, NJ: IEEE, 2020. ISBN 9781728196749. eISBN 9781728196732. p. 1-6. DOI: 10.1109/3ICT51146.2020.9312014. [Scopus; INSPEC] [FOR: N 009] [Input: 0,500].
4. **Mahmood, Z.** Computer aided design of block chain system and effectiveness analysis of smart systems for information integration // 10th international workshop on data analysis methods for software systems, Druskininkai, Lithuania, November 29 - December 1, 2018. Vilnius: Vilnius University press, 2018. ISBN 9786090700433. p. 56. [FOR: N 009]

Acknowledgements

I would like to take this opportunity to express my deepest gratitude and appreciation to the individuals who have been instrumental in my Ph.D. journey, making this remarkable achievement possible.

First and foremost, I am eternally grateful to my God, Lord Jesus Christ for his blessings, he who give me wisdom, enlightenment, and courage to face challenges, throughout my academic pursuit. His grace was a persistent source of strength to face and stood firm in every challenge of this journey.

I would like to extend my heartfelt thanks to my supervisor, Prof. Vacius Jusas, for his invaluable mentorship. His expertise, dedication, support, guidance, and encouragement have been key factors in shaping my personal and academic development. I am truly grateful for his surprising unwavering belief in my capabilities and for his tireless efforts to drive me to reach my full potential. I want to emphasize that without the firm support and guidance of my supervisor, this significant achievement would not have been possible. His profound insights, constructive feedback, and commitment to my growth have played a pivotal role in my research accomplishments.

I would like to express my gratitude to the KTU-INFORMATICS Department and Doctoral Committee for their expertise, valuable insights, and constructive feedback at various stages of my Ph.D. Their guidance and rigorous evaluation have been instrumental in refining my research and ensuring its academic rigor. I am truly appreciative of their commitment to academic excellence and their contribution to shaping my research work.

I would like to express my gratitude to the KTU-PHD school for its assistance with administrative tasks, Scholarship application, Conferences registration and making sure everything related to my doctoral studies was completed on time and correctly.

UDK 004.89+004.932](043.3)

SL 344. 2023-*, *, * leidyb. apsk. I. Tiražas 14 egz. Užsakymas * .
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas