Volume 10, Issue 3, March– 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar342

Signature Verification Using Deep Learning and CNN

G. Abinesh[1]; Dr. V. Kavitha[2]; Prajith. J. V[3]

(Associate Professor)[1]
[1,2,3]Department of Computer Science with Cognitive Systems, Sri Ramakrishna College of Arts and Science, Coimbatore.

Publication Date: 2025/03/19

Abstract: Signature verification plays a crucial role in authentication and fraud detection across various domains such as banking, legal documentation, and digital security. Traditional methods often struggle with intra-class variability, making deep learning approaches, particularly Convolutional Neural Networks (CNNs), a promising alternative. This study presents a CNN-based signature verification system that effectively distinguishes between genuine and forged signatures. The proposed model extracts spatial features from handwritten signatures using multiple convolutional layers, enabling robust feature learning. A Siamese network architecture is employed to compare signature pairs, utilizing contrastive or triplet loss to enhance verification accuracy. The system is trained on publicly available signature datasets and evaluated using performance metrics such as accuracy, precision, recall, and F1-score. Experimental results demonstrate that the CNN-based approach outperforms traditional feature-based methods, providing improved generalization to unseen signatures. This research highlights the potential of deep learning in enhancing signature verification reliability while reducing manual effort in forensic analysis. Index terms: Signature Verification, Convolutional Neural Networks, Deep Learning, Siamese Network, Authentication.

How to Cite: G. Abinesh; Dr. V. Kavitha; Prajith. J. V (2025). Signature Verification Using Deep Learning and CNN. *International Journal of Innovative Science and Research Technology*, 10(3), 374-381. https://doi.org/10.38124/ijisrt/25mar342

I. INTRODUCTION

Signature verification is a crucial process used to authenticate an individual's identity by analyzing and confirming the legitimacy of their handwritten or electronic signature. It plays a fundamental role in various sectors, including banking, legal transactions, business agreements, government records, and digital security, where ensuring the authenticity of a signature is vital in preventing fraud and unauthorized access. The verification process can be conducted manually by experts trained in handwriting analysis or through automated systems that use advanced technologies like machine learning, artificial intelligence, and biometric authentication to enhance accuracy and efficiency. With the increasing reliance on digital platforms, signature verification has evolved significantly from traditional methods to modern electronic and biometric techniques. Traditional verification methods involve comparing a signature against a known sample, assessing factors such as stroke patterns, pressure, spacing, and overall consistency. Expert forensic document examiners often perform this task in legal cases or financial disputes where signature forgery is suspected.

II. TYPES OF SIGNATURE VERIFICATION

A signature is a unique and personal identifier that can be analyzed based on several key features, including stroke pressure, signing speed, stroke order, and overall structure. Signature verification systems are designed to analyze and compare these attributes to determine the authenticity of a signature. The system can be broadly classified into two types: static (offline) verification and dynamic (online) verification. Static verification analyzes scanned or printed signatures, while dynamic verification captures real-time signing behavior, including pressure sensitivity and movement dynamics. These techniques help in distinguishing genuine signatures from forgeries, ensuring security and authenticity.

- *Data Retrieval*

The process of signature verification involves multiple stages, starting with data retrieval. Signature samples are extracted from image files or digital input devices, ensuring consistency and format standardization. Error handling mechanisms are implemented to detect inconsistencies in input data, such as distorted or incomplete signatures. Secure databases are utilized to store reference samples, providing a robust foundation for comparison. By incorporating both offline and real-time verification methods, the system offers a versatile approach to authentication.

- *Data Processing*

Data processing is another essential aspect of signature verification. Once a signature is retrieved, it undergoes pre-processing, which includes conversion to grayscale, noise reduction, and image enhancement. These techniques

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar342

improve the clarity of the signature, making it easier to extract crucial features. Feature extraction methods such as Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG), and Local Binary Patterns (LBP) are applied to identify distinctive characteristics within the signature. These features are then normalized to ensure consistency across different samples and conditions.

➢ *User Authentication*

User authentication is integrated into the system to provide additional security. Before submitting a signature for verification, users must be authenticated through secure login mechanisms. This prevents unauthorized access and ensures that the submitted signature corresponds to a registered individual. Reference signatures are stored in encrypted formats, adding an extra layer of security to the verification process.

➢ *Verification Algorithms*

The core of the signature verification system lies in its verification algorithms. Convolutional Neural Networks (CNNs) and other machine learning models play a crucial role in comparing signatures. These models are trained using large datasets of genuine and forged signatures, allowing them to learn patterns and detect discrepancies effectively. The comparison process involves measuring similarity using classifiers and distance metrics, which help determine the likelihood of a match. Integrating AI-driven techniques significantly enhances the system's ability to differentiate between authentic and fraudulent signatures with high precision.

➢ *User Interface (Gui)*

To ensure ease of use, the system is equipped with a graphical user interface (GUI) that allows users to upload signatures and receive instant verification results. The interface provides real-time feedback, highlighting key verification metrics and authentication confidence scores. Users can also configure settings such as verification thresholds and preferred authentication methods, allowing for a customizable experience. The GUI is designed to be intuitive and accessible, catering to both individual users and enterprise-level applications.

➢ *Database Design*

A well-structured database is essential for efficient signature verification. The system employs an optimized database architecture that stores signature samples along with user information. Indexing techniques are used to enable quick retrieval and matching, ensuring fast processing times. Security measures, including encryption and access controls, are implemented to protect sensitive data and prevent unauthorized modifications. Logging mechanisms are also integrated to maintain records of verification attempts, providing valuable insights for auditing and analysis.

➢ *System Architecture*

The system follows a layered architecture, consisting of three primary components:

- UI Layer: Facilitates user interaction, allowing users to input and retrieve verification results.
- Application Layer: Processes and analyzes signatures using machine learning models and feature extraction techniques.
- Data Layer: Manages storage, retrieval, and security of signature records.

III. TESTING AND IMPLEMENTATION

Thorough testing and implementation strategies are crucial for ensuring the reliability of the signature verification system. Unit testing is performed on individual modules, such as data extraction, feature processing, and authentication algorithms, to validate their functionality. Integration testing is conducted to ensure seamless interaction between different components of the system. Additionally, datasets containing both genuine and forged signatures are used to evaluate the system's accuracy and robustness. By refining algorithms and improving data processing techniques, the system continues to evolve and improve over time.

➢ *Impact of Artificial Intelligence on Signature Verification*

Artificial intelligence has revolutionized signature verification by enhancing the precision and efficiency of verification systems. AI-driven models, such as Convolutional Neural Networks (CNNs), employ deep learning techniques to analyze vast amounts of signature data and identify discrepancies. These models continuously learn from new data, improving their accuracy over time. AI-powered verification systems can quickly distinguish between genuine and fraudulent signatures, reducing manual workload and enhancing security.

➢ *Security Measures in Signature Verification*

Ensuring the security of signature verification systems is crucial in preventing unauthorized access and fraud. Multi-factor authentication (MFA) is commonly integrated with signature verification to provide an additional layer of security. Encryption techniques safeguard stored signature data, preventing unauthorized alterations. Furthermore, blockchain technology is being explored to create immutable signature records, ensuring authenticity and reducing the risk of forgery.

➢ *Ethical and Legal Implications*

Signature verification must adhere to ethical and legal guidelines to protect user privacy and data integrity. Regulatory compliance with frameworks like GDPR and data protection laws ensures that user data is handled securely and transparently. Ethical concerns regarding AI biases and signature data misuse must also be addressed through responsible AI practices and strict data governance policies.

➢ *Forecasting*

Forecasting plays a significant role in improving verification accuracy by adapting to evolving signature variations. Signatures naturally change over time due to aging, health conditions, and other factors, which can lead to false rejections in verification systems. Predictive analytics can help adjust verification thresholds dynamically, ensuring that

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar342

genuine signature variations are accurately identified while maintaining strict security measures against forgeries. This adaptability enhances user experience while ensuring the integrity of the verification system. Another aspect of forecasting in signature verification involves anticipating technological advancements and emerging threats. Cybersecurity risks, including deepfake signature attacks and AI-generated forgeries, pose significant challenges to traditional verification methods.

By leveraging forecasting models, researchers and developers can anticipate these threats and develop more sophisticated defense mechanisms, such as multimodal biometric authentication and blockchain-integrated verification systems. Predictive analysis also aids in resource allocation, allowing organizations to optimize their verification infrastructure based on anticipated demand and potential risks.

IV. DOCUMENTATION

Proper documentation in signature verification ensures transparency, reliability, and compliance with security standards. Comprehensive documentation serves as a reference guide for developers, auditors, security analysts, and regulatory bodies, facilitating smooth system operation, troubleshooting, and compliance verification. One of the primary components of documentation in signature verification is algorithmic workflows. Detailed records of signature verification algorithms, including feature extraction techniques, classification models, and verification thresholds, are essential for system maintenance and upgrades. These records help developers understand the underlying logic of the verification system, enabling them to refine algorithms and improve accuracy over time. Additionally, well-documented workflows facilitate seamless integration with emerging technologies such as AI-powered fraud detection and biometric authentication.

Verification logs form another critical aspect of documentation. Maintaining comprehensive logs of all verification attempts allows organizations to track authentication history, detect anomalies, and analyze user behavior. These logs include timestamped records of signature submissions, verification outcomes, confidence scores, and flagged discrepancies. In case of disputes or security breaches, verification logs serve as crucial evidence for auditing and forensic investigations.

Moreover, log data can be utilized for machine learning training, enabling continuous improvement of verification models.

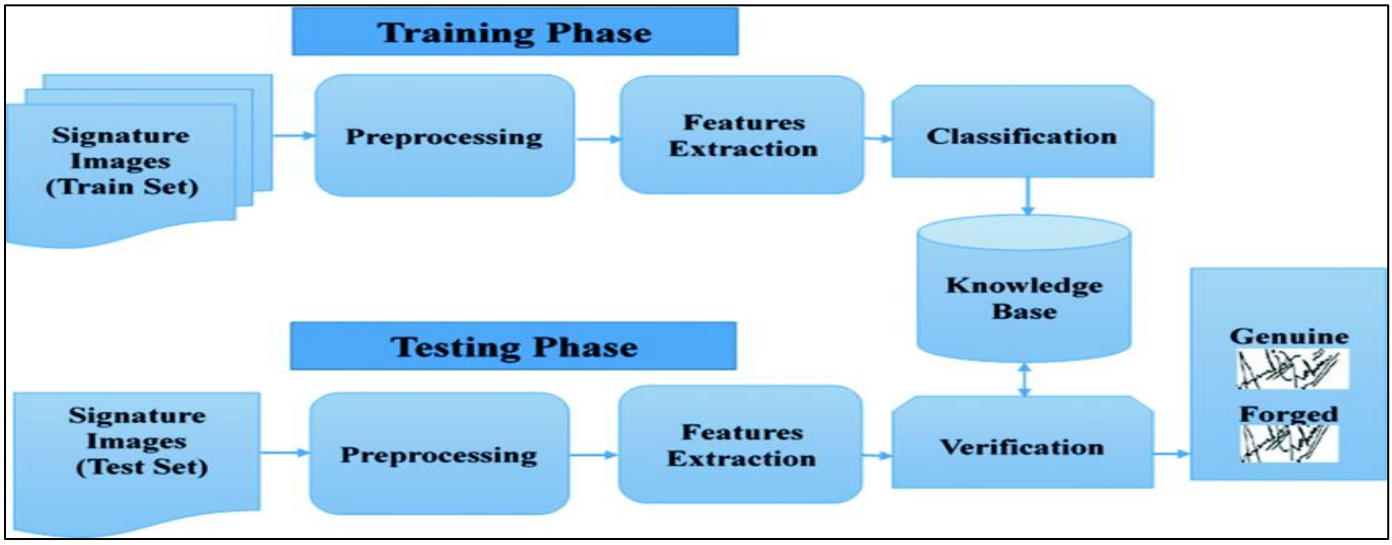

Fig 1 Training and Testing Phases of the CNN-Based Signature Verification System.

➢ *System Design*

- Command-Line Interface (CLI) Design: The system design for signature verification includes a user-friendly command-line interface (CLI) that guides users through the verification process efficiently.

- Main Menu: Upon launching the signature verification system, users are presented with a main menu containing options such as:

✓ Verify Signature
✓ Upload Reference Signature
✓ Set Verification Parameters
✓ Exit

- Signature Verification**:** When the user selects the "Verify Signature" option, they are prompted to upload a scanned or digitally captured signature. The system then compares this signature with stored reference samples, analyzing features such as stroke consistency, pressure sensitivity, and shape alignment to determine authenticity. The result is displayed, indicating whether the signature is genuine or potentially fraudulent.
- Upload Reference Signature: Users can upload reference signatures to build a trusted signature database. The system extracts essential features from the signature and securely stores them for future verification. This step is crucial for organizations requiring multi-user authentication.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar342

- Verification Parameters: Users can configure settings such as sensitivity levels for matching, error tolerance, and authentication thresholds. Adjusting these parameters allows flexibility based on security requirements and application domains.
- Error Handling: The system is designed to handle various error scenarios, including:

✓ Invalid or unreadable signature input
✓ Mismatched signature formats
✓ Database access errors
✓ Threshold mismatches leading to ambiguous verification results

V. GRAPHICAL USER INTERFACE (GUI) DESIGN

- Main Window: The GUI for signature verification is designed with a structured layout, allowing users to easily navigate through various functions. The main window contains sections for signature verification, uploading reference signatures, and accessing system settings. Clear, labeled buttons enable smooth interaction, making the process intuitive for users.
- Signature Upload and Verification Display: Users can upload signatures using an interactive drag-and-drop feature or by selecting a file manually. Once uploaded, the system processes the signature and visually displays verification results. The verification outcome, including match percentage and authenticity confidence level, is clearly presented with color-coded indicators to highlight genuine and fraudulent results.
- Reference Signature Database Management**:** A dedicated section in the GUI allows users to manage stored reference signatures. Users can add, update, or delete reference signatures as needed. This functionality ensures that organizations can maintain an up-to-date database for improved verification accuracy.
- Verification Parameters and Customization: The settings panel provides users with customization options, including the ability to adjust matching sensitivity, define acceptable signature variations, and enable advanced AI-based verification models. This feature allows users to tailor the verification process based on security requirements and specific application scenarios.
- User Feedback and Interaction: The GUI includes progress indicators during verification processes, ensuring users remain informed about ongoing operations. Interactive elements such as tooltips, hover effects, and instructional prompts enhance the usability of the system, reducing errors and improving the overall user experience.
- Error Handling and Notifications**:** The system is designed to handle and display errors effectively. Notifications appear in a dedicated message area, providing detailed descriptions of issues such as incorrect file formats, unreadable signatures, or database access failures. Color-coded alerts indicate the severity of errors, guiding users to take corrective actions promptly.

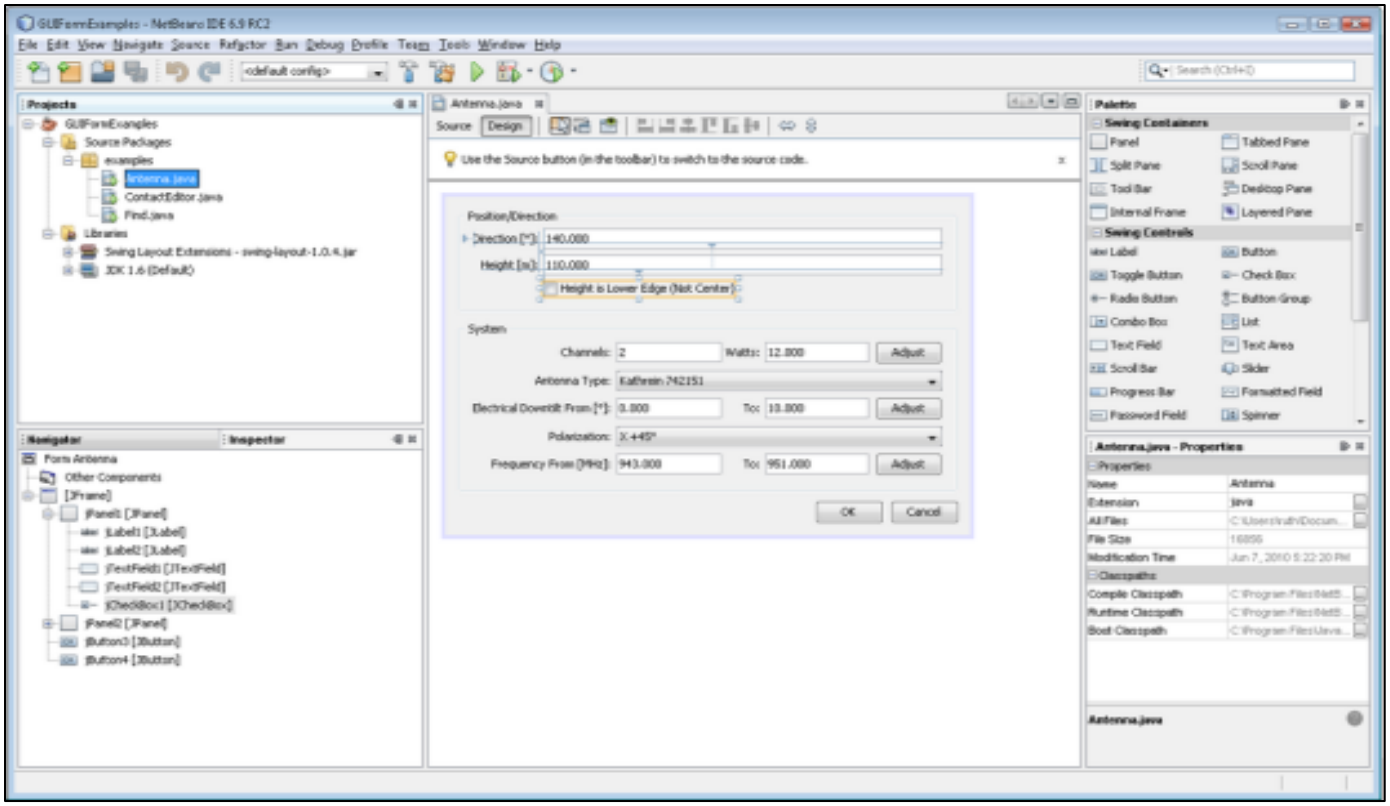

Fig 2 Software Interface for Model Configuration and Simulation

➢ *Signature Verification in the Database Context*

Signature verification can be employed in the database to ensure data integrity, authenticity, and security. It involves cryptographic techniques that allow verification of the data's origin and prevent unauthorized modifications. In the context of the Weather_Data table, signature verification

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar342

can ensure that recorded weather parameters remain unaltered from their original entry. This is particularly important when weather data is sourced from external APIs, meteorological agencies, or IoT-based weather sensors, as unauthorized alterations could mislead decision-makers and researchers.

➢ *Implementing Digital Signatures*

Digital signatures can be integrated within the SQLite database design using hashing and asymmetric encryption techniques. When a weather data record is inserted into the database, a digital signature can be generated using a private key. This signature can be stored as an additional column in the Weather_Data table. When a user retrieves or analyzes data, the corresponding signature is verified using the associated public key to ensure that the data has not been tampered with.

A sample schema modification for signature storage might be:

➢ *Alter Table Weather_Data Add Column Signature Text;*

- *Database Operations in Signature Verification*

Database operations play a crucial role in implementing and managing signature verification mechanisms. These operations include:

- Insertion of Signed Data: When a new weather record is added, the system generates a digital signature for the data using a private key and stores it alongside the record.
- Retrieval and Verification: When retrieving weather data, the system retrieves the stored digital signature and verifies it using the public key to ensure data integrity.
- Updating Records: If an update is necessary, a new signature must be generated to replace the previous one, ensuring the integrity of modified records.
- Deletion and Logging: Deleted records may be logged in an archive along with their signatures to maintain an auditable history of data changes.
- Batch Processing: Large datasets may require batch verification processes to ensure efficient integrity checking without excessive computational overhead.

VI. ROLE OF PUBLIC AND PRIVATE KEYS

In asymmetric cryptography, private keys are used to sign data, whereas public keys are used to verify the signatures. A meteorological agency or data provider could own the private key, ensuring that only their authorized systems can generate valid signatures. On the other hand, researchers, analysts, and decision-makers can use the public key to verify that the received data remains unaltered. This guarantees trust in the data's origin and integrity.

➢ *Preventing Unauthorized Modifications*

Without signature verification, unauthorized parties might alter weather records for various reasons, such as financial gain (e.g., modifying weather conditions to support fraudulent insurance claims) or misinformation campaigns. By enforcing digital signatures, any attempt to manipulate stored data would be detectable, as the modified record's hash would not match the original signed hash.

➢ *Timestamp Integrity*

The timestamp column in the Weather_Data table plays a vital role in verifying the accuracy of stored data. By ensuring that the timestamp is included in the signed data, signature verification can prevent attackers from modifying historical records without detection.

- This is especially crucial in applications where real-time weather tracking and predictions rely on accurate historical data.

➢ *Performance Considerations*

Although digital signatures enhance security, they introduce computational overhead. The process of generating, storing, and verifying signatures requires additional processing power and storage. However, SQLite is designed to handle lightweight database applications efficiently, and implementing digital signatures with optimized hashing algorithms ensures minimal performance impact.

➢ *System Architecture Flow in Signature Verification*

The system architecture for signature verification follows a structured flow that ensures the integrity, authenticity, and security of stored data. The architecture consists of several key components that work together to generate, store, and verify digital signatures efficiently.

- Data Acquisition Layer: The first step in the system architecture involves acquiring weather data from trusted sources such as meteorological sensors, third-party APIs, or manual entries by verified personnel. At this stage: The data is collected in real-time or batch mode. Each entry is formatted according to predefined standards. The collected data undergoes preliminary validation to check for anomalies.
- Signature Generation Module: Once the weather data is validated, a digital signature is generated using a cryptographic hashing algorithm combined with asymmetric encryption. The steps include: Computing a hash value for the data record (excluding the signature field). Encrypting the hash using a private key to produce a unique signature. Storing the signature alongside the weather record in the database.
- Database Storage Layer: The signed weather data is then stored securely in the SQLite database. The database operations for this layer include: Insertion of signed data to maintain data integrity. Updating records by generating new signatures for modified data. Logging deletion requests to prevent unauthorized removals.
- Data Retrieval and Verification Layer: When users or systems retrieve weather data, the signature verification process is initiated to ensure authenticity. The verification steps include: Retrieving the original data and its stored signature. Computing a new hash of the retrieved data. Decrypting the stored signature using the corresponding public key. Comparing the newly

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar342

computed hash with the decrypted hash; if they match, the data is verified as authentic.

- Access Control and Security Enforcement: To prevent unauthorized modifications and access to sensitive data: Role-based access controls (RBAC) are implemented. Only authorized personnel can modify data, and modifications trigger new signature generation. Logs of all access requests and modifications are maintained.
- Audit and Compliance Layer: To meet regulatory and security standards, the system includes: Regular integrity checks via batch signature verifications. Timestamp verification to ensure historical data remains intact. Secure logging of changes to maintain an audit trail.

VII. CONSIDERATIONS

Implementing signature verification requires attention to several key considerations to ensure robust security and efficiency:

➢ *Key Management*

Proper handling of cryptographic keys is essential. Private keys must be stored securely to prevent unauthorized access, while public keys should be easily accessible for verification. Key rotation policies should be implemented to mitigate risks of key compromise.

➢ *Performance Optimization*

Verifying digital signatures for large datasets can introduce computational overhead. To optimize performance:

- Batch verification techniques can be used for bulk data processing.
- Efficient hashing algorithms should be selected to balance security and speed.
- Indexing can be used in databases to improve retrieval times for signature verification.

➢ *Handling Data Updates*

Whenever data is modified, a new digital signature must be generated to reflect the changes. This ensures that altered records remain verifiable and trustworthy. However, excessive updates can lead to performance issues, requiring optimized update strategies.

➢ *Ensuring Backward Compatibility*

In cases where a signature verification system is added to an existing database, backward compatibility must be maintained. This can involve:

- Providing a fallback mechanism for older records without signatures.
- Implementing progressive verification where unsigned records are gradually updated with signatures.

➢ *Security Against Replay and Tampering Attacks*

Digital signatures should be timestamped to prevent replay attacks where old signatures are reused to forge authenticity. Additionally, cryptographic measures should be implemented to detect any unauthorized modifications to stored data.

➢ *Compliance with Regulatory Standards*

Different industries have varying compliance requirements regarding data integrity and security. Signature verification must align with standards such as:

- GDPR (General Data Protection Regulation)
- ISO/IEC 27001 (Information Security Management)
- NIST (National Institute of Standards and Technology) Guidelines

VIII. SYSTEM TESTING AND IMPLEMENTATION

➢ *Testing Methodologies:*

To ensure the effectiveness of signature verification, various testing methodologies should be employed:

- Unit Testing: Individual components such as hashing, encryption, and database storage are tested in isolation.
- Integration Testing: The interaction between different modules, including data acquisition, signature generation, and verification, is validated.
- Performance Testing: The system is tested for efficiency under varying loads to ensure that signature verification does not introduce excessive delays.
- Security Testing: Penetration testing and vulnerability assessments are conducted to identify potential security weaknesses.

➢ *Implementation Strategies*

The successful implementation of the signature verification system involves:

- Gradual Deployment: Rolling out the verification mechanism in phases to ensure smooth adoption.
- Data Migration: If existing records need to be signed, a migration strategy should be implemented to generate and store signatures for historical data.
- User Training: Ensuring that stakeholders understand the verification process and can interpret verification results correctly.
- Monitoring and Maintenance: Regularly monitoring system performance and updating cryptographic techniques to counter emerging threats.

➢ *Challenges and Mitigation Strategies*

- Computational Overhead: Optimization techniques such as indexing and batch processing can improve performance.
- Key Management: Secure storage and periodic rotation of cryptographic keys are essential.
- Backward Compatibility: Ensuring that older records without signatures can still be accessed and progressively signed over time.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar342

IX. CONCLUSION

➢ *Technological Innovations:*

Recent advancements in machine learning and deep learning have refined feature extraction and pattern recognition. These innovations enable systems to accommodate natural variations in handwriting while accurately distinguishing genuine signatures from forgeries. Integrating multimodal biometrics further strengthens overall system reliability.

- Persistent Challenges: Despite progress, issues such as intra-class variability and sophisticated forgery techniques continue to test the limits of current systems
- Operational Challenges: Despite progress, issues such as intra-class variability, sophisticated forgery techniques, and environmental inconsistencies continue to challenge system accuracy.

➢ *Continuous Research and Development Are Essential to Overcome These Limitations.*

- Pattern Recognition: Pattern recognition in signature verification refers to the process of analyzing and distinguishing key characteristics from a person's signature using advanced algorithms. These algorithms utilize machine learning and statistical techniques to identify consistent features such as stroke sequences, pressure variations, and overall structure. By examining these unique traits, the system can accurately identify whether a signature is genuine or forged. Leveraging pattern recognition allows for the detection of subtle discrepancies that may be missed by traditional methods, improving both accuracy and reliability. It plays a vital role in creating secure and robust verification systems for digital identity and transaction protection.
- Integrating multimodal biometrics: Integrating multimodal biometrics involves combining signature authentication with other biometric methods, like fingerprint or facial recognition, to strengthen security. By using multiple biometric factors, the system cross-checks various identifiers, reducing the risk of false acceptance or rejection that could occur with a single biometric trait. For example, if someone's signature appears to match but their fingerprint or facial features don't, the system flags the discrepancy, offering more reliable authentication.

➢ *Scope for Future Enhancement*

- *Improved Algorithms:*

Developing more robust and accurate algorithms, especially using deep learning, to handle variability in signatures and detect sophisticated forgeries.

- *Multimodal Biometric Integration:*

Combining signature verification with other biometric systems (e.g., fingerprints, facial recognition) to increase security and minimize fraud risk.

- *Adaptive Learning Models:*

Implementing systems that learn from new data and adapt to changes in an individual's signature over time, ensuring long-term reliability.

- *Real-Time Verification:*

Enhancing the speed and efficiency of signature verification systems for real-time applications in online banking, e-commerce, and secure transactions.

- *Cross-Platform Usability:*

Ensuring seamless signature verification across different devices and platforms, including smartphones, tablets, and PCs.

- *Forgery Detection:*

Developing more advanced techniques to detect sophisticated signature forgeries, including the use of AI-driven anomaly detection systems.

- *Cloud-Based Solutions:*

Expanding cloud-based signature verification to enable easy scalability, remote access, and global applications.

- *User Experience Improvements:*

Balancing security with a seamless, user-friendly experience, ensuring minimal friction in authentication processes.

REFERENCES

[1]. Chatzisterkotis, Thomas (2015) An examination of quantitative methods for Forensic Signature Analysis and the admissibility of signature verification system as legal evidence. Master of Science by Research (MScRes) thesis, University of Kent, https://kar.kent.ac.uk/id/eprint/54048.

[2]. Dr.V. Thangavel (2023) Use of Digital Signature Verification System (DSVS) in various Industries: Security to protect against counterfeiting: Research. Z-Global Banking eJournal Vol 2, Issue 15.

[3]. Syed Zulkarnain Syed Idrus, Estelle Cherrier, Christophe Rosenberger, Jean-Jacques Schwartzmann. (2013) A Review on Authentication Methods. Australian Journal of Basic and Applied Sciences, 7 (5), pp.95-107. hal-00912435.

[4]. Abhishek Shende, Mahidhar Mullapudi and Narayana Challa, (2024) Enhancing Document Verification Systems: A Review of Techniques, Challenges, and Practical Implementations, International Journal of Computer Engineering and Technology (IJCET),15(1),16-25. https://iaeme.com/Home/issue/IJCET?Volume=15&Issue=1.

[5]. N. Zaman, I. Karabey Aksakallı, and N. Bayğın, (2023) "Digital Certificate Security: A Blockchain-based Approach for Fraud Prevention and Verification", Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, vol. 12, no. 4, pp. 1128–1138, doi:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar342

10.17798/bitlisfen.1343747.Wang, C., Li, Q., & Kim, S. (Eds.).

[6]. Fierrez-Aguilar, J., Krawczyk, S., Ortega-Garcia, J., Jain, A.K., 2005b. Fusion of local and regional approaches for on-line signature verification. IWBRS 2005, 188–196.

[7]. Jain, A.K., Griess, F., Connell, S. (2002). On-line signature verification. Pattern Recogn. 35, 2963–2972.

[8]. Kashi, R.S., Hu, J., Nelson, W.L., Turin, W., (1997). On-line handwritten signature verification using Hidden Markov Model features. In: Proceedings of the ICDAR, pp. 253–257.

[9]. Krawczyk, S., (2005). User authentication using on-line signature and speech. Master's Thesis, Michigan State University, Department of Computer Science and Engineering.