**CHAPTER ONE**

**1.1 Background of the Study**

In the modern era of banking and finance, digital transactions have become increasingly prevalent due to their convenience and efficiency. However, with this growing reliance on digital systems, concerns about security and fraud have also heightened. One of the significant challenges in the digital banking domain is verifying the authenticity of transactions. To address this issue, the implementation of a robust system for verifying bank transactions using signatures has gained prominence. Traditional banking methods often rely on handwritten signatures as a means of identification and verification. As technology advances, integrating signature verification systems into digital banking platforms offers an opportunity to enhance security while maintaining the ease of conducting transactions for customers. This chapter delves into the development of such a system, analyzing its problem statement, objectives, significance, scope, limitations, and defining key terms to set the stage for the study.

**1.2 Problem Statement**

The problem at hand is that current digital banking systems often rely solely on passwords, PINs, or other authentication methods, which can be vulnerable to various forms of fraud and security breaches. These methods lack the robustness of signature-based verification, which has long been recognized as a secure means of identifying individuals in the physical world.Hence, the challenge lies in implementing a digital signature verification system that can reliably and accurately authenticate bank transactions, thereby bolstering security, reducing fraudulent activities, and providing customers with greater peace of mind.

**1.3 Aim and Objectives**

**1.3.1 Aim:**

The aim of this study is to develop and implement a system for verifying bank transactions using digital signatures to enhance security and reduce fraud.

**1.3.2 Objectives:**

To achieve the aim, the following specific objectives will be pursued:

1. Investigate the existing signature verification technologies and methodologies used in various industries.

2. Design and develop a robust signature capture mechanism for digital transactions.

3. Create an efficient algorithm for comparing and verifying digital signatures.

4. Integrate the signature verification system into the existing digital banking infrastructure.

5. Evaluate the system's performance in terms of accuracy, efficiency, and security.

6. Identify potential areas of improvement and propose recommendations for future enhancements.

**1.4 Significance of the Study**

The significance of this study lies in its potential to strengthen the security measures in digital banking systems and protect customers from unauthorized access and fraudulent transactions. By integrating signature verification into the digital realm, banks can instill greater confidence in their customers, leading to increased usage of digital services and improved customer loyalty.

Moreover, A successful implementation of this system would position banks at the forefront of innovation and security in the financial sector, setting them apart from their competitors and attracting tech-savvy customers.

**1.5 Scope and Limitations**

This study focuses on the development and implementation of a signature verification system for digital banking transactions. It encompasses the technical aspects of signature capture, processing, and comparison, as well as the integration of the system into existing banking infrastructure.

**Some of the potential limitations of this study include**:

**1. Availability of data:** The accuracy and performance of the signature verification system heavily depend on the availability of a diverse dataset for training and testing the algorithms. Obtaining a comprehensive dataset might be a challenge.

**2. Hardware constraints:** The success of the system may be influenced by the capabilities of the hardware devices used for signature capture. Compatibility issues with various devices could arise.

**3. Ethical considerations:** Privacy and data protection must be carefully considered throughout the development and implementation of the system to ensure compliance with legal and ethical standards.

**1.6 Definition of Terms**

**1. Digital Signature:** A cryptographic representation of a person's identity used to authenticate electronic documents and transactions.

**2. Verification:** The process of confirming the authenticity of a digital signature by comparing it with a reference signature.

**3. Algorithm:** A set of rules and procedures used to perform specific tasks, such as signature comparison in this study.

**4. Fraud:** Unauthorized and deceptive activities carried out with the intention of gaining financial or personal benefits illegally.

**5. Authentication:** The process of verifying the identity of a user or entity to grant access to secure systems or data.

With a clear understanding of the background, problem statement, objectives, significance, scope, limitations, and key terms, this study aims to pave the way for the development and implementation of a robust system for verifying bank transactions using digital signatures. By addressing the challenges in digital banking security, this system has the potential to enhance customer trust, protect financial interests, and contribute to the advancement of the banking industry.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Literature Review:**

Biometrics technology is used in a wide variety of security applications. The aim of such systems is to recognize a person based on physiological or behavioral traits. In the first case, the recognition is based on measurements of biological traits, such as the fingerprint, face, iris, etc. The later case is concerned with behavioral traits such as voice and the handwritten signature (Edson J. R. Justino et al). Biometric systems are mainly employed in two scenarios: verification and identification. In the first case, a user of the system claims an identity, and provides the biometric sample. The role of the verification system is to check if the user is indeed who he or she claims to be. In the identification case, a user provides a biometric sample, and the objective is to identify it among all users enrolled in the system. The handwritten signature is a particularly important type of biometric trait, mainly due to its ubiquitous use to verify a person’s identity in legal, financial and administrative areas. One of the reasons for its widespread use is that the process to collect handwritten signatures is non-invasive, and people are familiar with the use of signatures in their daily life [48].

Signature verification systems aim to automatically discriminate if the biometric sample is indeed of a claimed individual. In other words, they are used to classify query signatures as genuine or forgeries. Forgeries are commonly classified in three types: random, simple and skilled (or simulated) forgeries. In the case of random forgeries, the forger has no information about the user or his signature and uses his own signature instead. In this case, the forgery contains a different semantic meaning than the genuine signatures from the user, presenting a very different overall shape. In the case of simple forgeries, the forger has knowledge of the user’s name, but not about the user’s signature. In this case, the forgery may present more similarities to the genuine signature, in particular for users that sign with their full name, or part of it. In skilled forgeries, the forger has access for both the user’s name and signature, and often practices imitating the user’s signature. This result in forgeries that have higher resemblance to the genuine signature, and therefore are harder to detect. Depending on the acquisition method, signature verification systems are divided in two categories: online (dynamic) and offline (static). In the online case, an acquisition device, such as a digitizing table, is used to acquire the user’s signature. The data is collected as a sequence over time, containing the position of the pen, and in some cases including additional information such as the pen inclination, pressure, etc. In offline signature verification, the signature is acquired after the writing process is completed. In this case, the signature is represented as a digital image [31].

Over the last few decades, some key survey papers have summarized the advancements in the field, in the late 80’s [47], 90’s [40] and 2000’s [31]. Some recent advancements have been consolidated in more recent literature reviews: Impedovo et al. [32] provide an update over the authors’ previous review [31], focusing on advancements such as new acquisition devices (mostly for online signatures) and signature representations; Shah et al. [58] present a critical evaluation of 15 signature verification systems proposed in the literature, classifying each work according to the feature extraction methods, classifiers and overall strengths and limitations of the systems. These reviews, on the other hand, do not capture more recent trends in the field, in particular the usage of Deep Learning methods applied for handwritten signatures. Such methods have demonstrated superior results in multiple benchmarks, and are reviewed in the present work.

**2.1.1 Machine Learning System:**

A machine learning system is a subset of artificial intelligence (AI) that focuses on developing systems capable of learning and improving their performance based on the data they consume. Machine learning aims to construct computer programs that can learn from data and make predictions. It intersects with various scientific disciplines such as statistics, cognitive science, and information theory. Machine learning has been successfully applied to solve real-world problems, including spam email classification, fraud detection, and customer relationship management. Its effectiveness in making predictions and aiding decision-making processes has led to its widespread adoption by companies and non-profit organizations.

**2.1.2 Benefits of Machine Learning:**

Machine learning methods offer several benefits in computer science projects. They have been widely employed to solve complex problems and make accurate predictions. Machine learning has proven effective in tasks such as spam email classification, sentiment analysis, recommendation systems, and natural language processing. Its ability to learn patterns from data and adapt to changing conditions makes it a valuable tool in computer science project work. Machine learning also offers a promising career path and contributes to better decision-making in various domains.

**2.2 Component of a Machine Learning Model System:**

A machine learning model system consists of three key components: representation, evaluation, and optimization. These components are essential in both supervised and unsupervised learning. In the context of computer science projects, these components play a crucial role in developing and improving machine learning mode.

**2.2.1 Representation:**

Representation refers to how data is structured and represented for the machine learning model. It involves selecting appropriate features, preprocessing data, and transforming data into a suitable format for analysis. In computer science projects, representation choices may include considering individual data points or organizing data in a graph structure for more complex analyses.

**2.2.2 Evaluation:**

Evaluation involves assessing the performance of the machine learning model to determine its accuracy and effectiveness. In supervised learning, evaluation metrics such as accuracy, precision, recall, and F1-score are commonly used. In unsupervised learning, evaluation metrics such as silhouette coefficient, Dunn index, and Calinski-Harabasz index can measure the quality of clustering results. Evaluation metrics are crucial in determining the success of a machine learning model in crime prediction and other computer science applications.

**2.2.3 Optimization:**

Optimization focuses on finding the best model parameters or configuration to achieve optimal performance. Optimization techniques vary depending on the machine learning algorithm used. In computer science projects, optimization techniques such as grid search, random search, and Bayesian optimization are commonly employed to fine-tune models and improve their performance.

**2.3 Design Methodology:**

This section discusses different software engineering methodologies commonly used in computer science projects. These methodologies provide systematic approaches to manage and execute software engineering projects. In the context of crime prediction using k-means clustering, the following methodologies are relevant:

**2.3.1 Structured System Analysis and Design Methodology (SSADM):**

SSADM is a pragmatic system analysis and design methodology that involves investigating the present system, defining the new system, establishing constraints, and analyzing and documenting the system. In computer science projects, utilizing SSADM can help in developing system flow charts, job steps, and program narratives, which enhance organization and facilitate computer execution.

**2.3.2 Prototyping:**

Prototyping is a software development methodology that allows developers to create a preliminary version of the solution to demonstrate functionality and gather feedback. In computer science projects, prototyping can be used to validate the feasibility of crime prediction using k-means clustering and refine the solution based on user feedback.

**2.4.3 Techniques Used in Analysis**

**I. Data Collection and Preprocessing**

A substantial dataset of signature samples from account holders is collected to build the prediction models. Signature samples are collected from bank customers with their consent. The dataset includes genuine signatures for positive samples and randomly generated signatures for negative samples to create a balanced dataset. Preprocessing techniques, like image normalization and noise removal, are applied to ensure data quality. Data collection for the Bank Verification System involves the collaboration of banks and their customers. Customers willingly provide their signatures, acknowledging that this process is an essential step in enhancing security measures. The dataset comprises two types of samples: genuine signatures from authorized account holders (positive samples) and randomly generated signatures from individuals without authorized access (negative samples). By having both positive and negative samples, we create a balanced dataset that aids in training the prediction models effectively. To ensure data quality and consistency, various preprocessing techniques are applied. Image normalization is utilized to standardize the signatures' size and orientation, making them comparable for feature extraction and model training. This step eliminates potential discrepancies caused by variations in signature sizes or alignment. Another crucial preprocessing step is noise removal. Signatures may contain imperfections or irrelevant details that can hinder model accuracy. Noise removal techniques, such as Gaussian blurring or median filtering, are employed to smooth the signature images while retaining their essential characteristics.

Additionally, outlier detection is implemented during the data preprocessing phase. Outliers, which may arise from inaccurate signature captures or anomalous samples, are identified and removed to prevent their influence on the model's learning process.

Data augmentation techniques are also considered to augment the dataset's size and diversity. By applying slight variations to the genuine signatures, such as rotation, scaling, and translation, we can enhance the model's ability to generalize and recognize signatures with different styles.

The preprocessed dataset is then split into training, validation, and testing sets. The training set is used to train the prediction models, while the validation set helps fine-tune the model's hyperparameters and avoid overfitting. The testing set is used to evaluate the final model's performance on unseen data, providing valuable insights into its real-world effectiveness. Careful attention is given to the privacy and security of customer signatures throughout the data collection and preprocessing stages. All personal information is anonymized and encrypted to prevent any potential data breaches or misuse.

By implementing comprehensive data collection and preprocessing strategies, the Bank Verification System ensures that the prediction models receive high-quality data for effective signature verification. This rigorous approach not only enhances the accuracy and reliability of the system but also instills confidence among both banks and their customers in the overall security of their financial transactions.

As with most pattern recognition problems, preprocessing plays an important role in signature verification. Signature images may present variations in terms of pen thickness, scale, rotation, etc., even among authentic signatures of a person. Below we summarize the main preprocessing techniques:

• **Signature extraction**: This is an initial step that consists in finding and extracting a signature from a document. This is a particular challenging problem in bank cheques, where the signature is often written on top of a complex background Giovanni Dimauro et al.,, S. Djeziri et al.,. This step is, however, not considered in most signature verification studies, that already consider signatures extracted from the documents.

• **Noise Removal:** Scanned signature images often contain noise. A common strategy to address this problem is to apply a noise removal filter to the image, such a median filter Kai Huang et al.,. It is also common to apply morphological operations to fill small holes and remove small regions of connected components Kai Huang et al., Mustafa Berkay et al.,.

**• Size normalization and centering**: Depending on the properties of the features to be used, different size normalization strategies are adopted. The simplest strategy is to crop the signature images to have a tight box on the signature S. Ghandali and M.E. Moghaddam. Another strategy is to user a narrower bounding box, such as cutting strokes that are far from the image centroid, that are often subject to more variance in a user’s signature Mustafa Berkay et al.,. Other authors use a fixed frame size (width and height), and center the signature in this frame M. Pourshahabi et al.,, [Luiz G. Hafemann et al.,].

II. Offline signature verification has been studied from many perspectives, yielding multiple alternatives for feature extraction. Broadly speaking, the feature extraction techniques can be classified as Static or Pseudo-dynamic, where pseudo dynamic features attempt to recover dynamic information from the signature execution process (such as speed, pressure, etc.). Another broad categorization of the feature extraction methods is between Global and Local features. Global features describe the signature images as a whole - for example, features such as height, width of the signature, or in general feature extractors that are applied to the entire signature image. In contrast, local features describe parts of the images, either by segmenting the image (e.g. according to connected components) or most commonly by the dividing the image in a grid (of Cartesian or polar coordinates), and applying feature extractors in each part of the image. Recent studies approach the problem from a representation learning perspective [Luiz G. Hafemann et.,], [Luiz G. Hafemann et al.,], [H. Rantzsch et al], [Z. Zhang et al ]: instead of designing feature extractors for the task, these methods rely on learning feature representations directly from signature images. A. Handcrafted feature extractors A large part of the research efforts on the field has been devoted to finding good feature representations for offline signatures. In this section we summarize the main descriptors proposed for the problem. 1) Geometric Features: Geometric features measure the overall shape of a signature. This includes basic descriptors, such as the signature height, width, caliber (height-to-width ration) and area. More complex descriptors include the count of endpoints and closed loops H. Baltzakis and N. Papamarkos. Besides using global descriptors, several authors also generate local geometric features by dividing the signature in a grid and calculating features from each cell. For example, using the pixel density within grids H. Baltzakis and N. Papamarkos, A. El-Yacoubi et al, [Edson J. R. Justino]. 2) Graphometric features: Forensic document examiners use the concepts of graphology and graphometry to examine handwriting for several purposes, including detecting authenticity and forgery. Oliveira et al. [Luiz S. Oliveira] investigated applying such features for automated signature verification. They selected a subset of graphometric features that could be described algorithmically, and proposed a set of feature descriptors. They considered the following static features: Calibre - the ratio of Height / Width of the image; Proportion, referring to the symmetry of the signature, Alignment to baseline - describing the angular displacement to an horizontal baseline, and Spacing - describing empty spaces between strokes.

• **Signature representation** - Besides just using the gray level image as input to the feature extractors, other representations have been considered. For instance, using the signature’s skeleton, outline, ink distribution, high pressure regions and directional frontiers Kai Huang et al.,.

**2. Feature Extraction:**

Relevant features are extracted from the signature images to create numerical representations suitable for modeling. Offline signature verification has been studied from many perspectives, yielding multiple alternatives for feature extraction. Broadly speaking, the feature extraction techniques can be classified as Static or Pseudo-dynamic, where pseudo dynamic features attempt to recover dynamic information from the signature execution process (such as speed, pressure, etc.). Another broad categorization of the feature extraction methods is between Global and Local features. Global features describe the signature images as a whole - for example, features such as height, width of the signature, or in general feature extractors that are applied to the entire signature image. In contrast, local features describe parts of the images, either by segmenting the image (e.g. according to connected components) or most commonly by the dividing the image in a grid (of Cartesian or polar coordinates), and applying feature extractors in each part of the image. Recent studies approach the problem from a representation learning perspective [Luiz G. Hafemann et.,], [Luiz G. Hafemann et al.,], [H. Rantzsch et al], [Z. Zhang et al ]: instead of designing feature extractors for the task, these methods rely on learning feature representations directly from signature images. A. Handcrafted feature extractors A large part of the research efforts on the field has been devoted to finding good feature representations for offline signatures. In this section we summarize the main descriptors proposed for the problem. 1) Geometric Features: Geometric features measure the overall shape of a signature. This includes basic descriptors, such as the signature height, width, caliber (height-to-width ration) and area. More complex descriptors include the count of endpoints and closed loops H. Baltzakis and N. Papamarkos. Besides using global descriptors, several authors also generate local geometric features by dividing the signature in a grid and calculating features from each cell. For example, using the pixel density within grids H. Baltzakis and N. Papamarkos, A. El-Yacoubi et al, [Edson J. R. Justino]. 2) Graphometric features: Forensic document examiners use the concepts of graphology and graphometry to examine handwriting for several purposes, including detecting authenticity and forgery. Oliveira et al. [Luiz S. Oliveira] investigated applying such features for automated signature verification. They selected a subset of graphometric features that could be described algorithmically, and proposed a set of feature descriptors. They considered the following static features: Calibre - the ratio of Height / Width of the image; Proportion, referring to the symmetry of the signature, Alignment to baseline - describing the angular displacement to an horizontal baseline, and Spacing - describing empty spaces between strokes.

**2.4 Prediction Models:**

Bank Verification System that utilizes signature verification as a means of enhancing security in banking transactions. Signatures have been widely used as a form of personal identification, and leveraging this unique characteristic can add an extra layer of authentication to prevent fraudulent activities. The proposed system aims to develop and evaluate two prediction models to authenticate signatures and ensure the integrity of banking operations. This paper outlines the methodology, data collection, and model implementation, followed by a comprehensive evaluation of the prediction models' performance. The results demonstrate the efficacy of the Bank Verification System using signature-based authentication, providing valuable insights for banks and financial institutions seeking to bolster their security protocols.

**3.Model Selection:**

The success of the Bank Verification System heavily relies on the selection of appropriate prediction models for signature verification. After careful consideration of various machine learning algorithms, two powerful models, Support Vector Machines (SVM) and Convolutional Neural Networks (CNN), are chosen for this critical task.

**1. Support Vector Machines (SVM):**

SVM is a widely-used supervised learning algorithm that excels in binary classification tasks. It works by finding the optimal hyperplane that best separates data points belonging to different classes. In the context of signature verification, SVM is particularly effective because of its ability to handle high-dimensional feature spaces and its robustness against overfitting. For SVM model training, the dataset of preprocessed signature features is used. The positive samples (genuine signatures) are assigned a label of "1," and the negative samples (randomly generated signatures) are assigned a label of "0." SVM then learns to create a decision boundary that maximizes the margin between the two classes while minimizing classification errors. This ensures that the SVM model can effectively distinguish between genuine and forged signatures.

**2. Convolutional Neural Networks (CNN):**

CNN is a type of deep learning model designed specifically for image-related tasks. It has demonstrated outstanding performance in image classification and recognition tasks, making it an ideal candidate for signature verification, which involves analyzing image-based signatures. In CNN, the input data is processed through multiple layers of convolutions, followed by pooling and fully connected layers. Each convolutional layer extracts relevant features from the signature images, and the pooling layers down sample the features to reduce computation complexity. The fully connected layers then use the extracted features to make a final prediction.

To train the CNN model, the signature images are used as input. The CNN model learns to recognize unique patterns, strokes, and distinctive features present in genuine signatures. The training process involves minimizing the loss function and updating the model's weights to improve its performance. The powerful feature extraction capabilities of CNN enable it to effectively differentiate between genuine and forged signatures, even amidst variations in writing styles.

**2.5. Model Evaluation:**

Both the SVM and CNN models undergo rigorous evaluation to determine their effectiveness in signature verification. The preprocessed testing set, which contains unseen signature samples, is used to assess the models' performance. Various evaluation metrics are used, including accuracy, precision, recall, and F1 score, to quantify the models' predictive capabilities. Accuracy measures the overall correctness of predictions, while precision and recall provide insights into the models' abilities to correctly identify positive samples (genuine signatures) and negative samples (forged signatures), respectively. The F1 score offers a balanced measure of the models' precision and recall.

Various evaluation metrics are employed to quantitatively measure the models' predictive capabilities and provide a complete understanding of their performance:

**1. Accuracy:**

Accuracy is a fundamental evaluation metric that measures the overall correctness of predictions made by the models. It represents the proportion of correctly classified samples (both genuine and forged signatures) to the total number of samples in the testing set. A higher accuracy value indicates that the models can effectively discriminate between genuine and forged signatures.

**2. Precision:**

Precision assesses the models' ability to correctly identify positive samples, which in this case are genuine signatures. It measures the ratio of true positive predictions (correctly identified genuine signatures) to the total number of samples predicted as genuine. A high precision score indicates that the models are reliable in recognizing genuine signatures without misclassifying them as forged.

**3. Recall:**

Recall, also known as sensitivity or true positive rate, gauges the models' sensitivity to correctly identify positive samples (genuine signatures) out of all actual positive samples in the testing set. It measures the ratio of true positive predictions to the total number of actual genuine signatures. A high recall value indicates that the models can effectively capture a significant portion of genuine signatures without missing them.

**4. F1 Score:**

The F1 score is a balanced measure that combines both precision and recall. It represents the harmonic mean of precision and recall and is calculated as (2 \* Precision \* Recall) / (Precision + Recall). The F1 score provides a single metric to assess the overall performance of the models, taking into account both the correct classification of genuine signatures and the minimization of false negatives. By evaluating the SVM and CNN models using these metrics, we gain a comprehensive understanding of their strengths and weaknesses in signature verification. High accuracy, precision, recall, and F1 score values signify that the models are effective in distinguishing genuine signatures from forged ones, providing reliable security measures for the Bank Verification System. The rigorous evaluation of the models ensures that the Bank Verification System is equipped with trustworthy and robust signature verification capabilities. These results instill confidence among banks, customers, and financial institutions, as the system demonstrates its ability to safeguard transactions and prevent unauthorized access, bolstering the security of the banking sector. Moreover, this evaluation process allows for continuous improvement and optimization of the models to adapt to evolving security challenges and maintain the system's effectiveness over time.

**4. Comparison and Optimization:**

The evaluation results are thoroughly analyzed to understand each model's strengths and weaknesses. The performance of SVM and CNN is compared to identify the most effective model for signature verification in the Bank Verification System. Additionally, hyperparameter tuning is performed to optimize the models and achieve their peak performance. By carefully selecting and fine-tuning the prediction models, the Bank Verification System ensures that it can reliably and accurately authenticate customer signatures, thereby enhancing security and mitigating potential fraudulent activities. The combination of SVM's robustness and CNN's feature extraction capabilities provides a powerful and effective solution for safeguarding banking transactions and building trust among customers and financial institutions.

**2.5 Model Training:**

Model training is a critical phase in the development of the Bank Verification System using signature-based authentication. In this phase, the selected prediction models, Support Vector Machines (SVM) and Convolutional Neural Networks (CNN), are trained using the extracted features from the preprocessed signature dataset. The goal of model training is to enable the models to learn and recognize the unique patterns and characteristics present in genuine signatures, thus enabling them to accurately distinguish between genuine and forged signatures.

**1. SVM Model Training:**

For training the SVM model, the preprocessed dataset is split into two subsets: the training set and the validation set. The training set comprises a significant portion of the dataset, and it is used to teach the SVM model to create the optimal decision boundary that separates genuine signatures from forged signatures. The SVM model learns to map the extracted features of genuine signatures to a positive class (labeled as "1") and the features of forged signatures to a negative class (labeled as "0"). During the training process, the SVM algorithm iteratively updates its parameters to find the hyperplane that maximizes the margin between the two classes while minimizing misclassifications. To avoid overfitting, the SVM model's hyperparameters, such as the regularization parameter (C) and the kernel type, are fine-tuned using the validation set. The model with the best performance on the validation set is selected for final testing and evaluation.

**2. CNN Model Training:**

CNN model training involves using the preprocessed signature images as input data. Similar to SVM, the dataset is divided into training and validation sets. The CNN model learns to recognize the complex and intricate patterns present in genuine signatures by processing the input images through a series of convolutional and pooling layers. During the training process, the CNN model's weights are iteratively updated using backpropagation to minimize the loss function. The objective is to improve the model's ability to accurately classify genuine signatures while disregarding irrelevant variations. To prevent overfitting, the CNN model may include regularization techniques such as dropout or L2 regularization. These techniques help in generalizing the model's learning to unseen signatures, making it more robust in real-world scenarios.

**3. Optimization and Performance Metrics:**

Throughout the training process, the models' performance is regularly monitored using metrics such as accuracy, precision, recall, and F1 score. These metrics provide insights into how well the models are learning to differentiate between genuine and forged signatures. The optimization of both models involves iteratively adjusting the model's hyperparameters, architecture, or optimization algorithms to achieve better performance. The goal is to find the right balance between model complexity and generalization ability.

**2.7. Final Model Selection and Testing:**

Once the SVM and CNN models are trained and optimized, the final models are selected based on their performance on the validation set. The selected models are then evaluated using the independent testing set, which contains signature samples that were not used during the training or validation stages. The testing set evaluates the models' ability to generalize to unseen data and provides a realistic assessment of the Bank Verification System's performance in real-world scenarios. The models with the highest accuracy and the best overall performance on the testing set are chosen for deployment in the Bank Verification System. By effectively training and optimizing the SVM and CNN models, the Bank Verification System achieves high accuracy and reliability in signature verification, significantly enhancing security and preventing fraudulent activities in the banking sector. The combination of both models' strengths allows for a robust and efficient authentication process, instilling confidence in customers and financial institutions alike.

**2.6. Model Implementation:**

The SVM and CNN models are implemented using standard libraries and frameworks. SVM is chosen for its simplicity and effectiveness in binary classification tasks, while CNN's ability to learn complex patterns makes it suitable for signature verification.

**2.7. Model Evaluation:**

The performance of the SVM and CNN models is assessed using metrics such as accuracy, precision, recall, and F1 score. A comparison of the models' performance is presented to identify the most effective prediction model for signature verification.

**2.8. Results and Discussion:**

The evaluation results show that both SVM and CNN achieve high accuracy in signature verification, with CNN exhibiting better performance due to its ability to capture intricate signature patterns. The Bank Verification System using signature authentication demonstrates its effectiveness in enhancing security and preventing fraudulent activities.

**2.9 Summary**

This chapter concludes with a summary of the key findings from the literature review. The insights gained from this review will inform the design and development of the system for verifying bank transactions using signatures, ensuring its alignment with the latest advancements and best practices in the field. By building on the existing body of knowledge, the proposed system aims to contribute to the enhancement of digital banking security and customer trust.

**CHAPTER THREE**

**METHODOLOGY**

**3.1 Requirements Analysis**

Before diving into the technical details of the system, it is essential to conduct a thorough requirements analysis to determine the needs and expectations of our crime prediction system's stakeholders. This analysis involves gathering information from our case study, commercial establishments, and other relevant parties to understand their specific requirements for bank verification using signature. The collected requirements will serve as the foundation for the subsequent stages of the system development process.

**3.2 Bank Verification System using Support Vector Machine Model**

In this section, we will discuss the analysis of the existing bank verification system and propose a redesigned system that utilizes the Support Vector Machine (SVM) model for enhanced security and accuracy in signature verification.

**3.2.1 Analysis of Existing System:**

The current bank verification system relies on traditional methods of user identification, such as PINs and passwords. While these methods have been widely used, they are susceptible to security breaches and fraudulent activities. Unauthorized access to accounts and identity theft continue to pose significant challenges for the banking sector. One of the main limitations of the existing system is its vulnerability to attacks, as PINs and passwords can be easily compromised or guessed by malicious actors. Moreover, the system lacks an additional layer of security to ensure that the person conducting the transaction is the legitimate account holder.

**3.2.2 Design Improvements using Support Vector Machine Model:**

To address the limitations of the existing system, we propose a redesigned bank verification system that leverages the power of the Support Vector Machine (SVM) model for signature verification. The SVM model is a robust machine learning algorithm that is highly effective in binary classification tasks, making it suitable for distinguishing between genuine and forged signatures.

The redesigned system's key components and workflow are as follows:

**1. Signature Data Collection:**

Customers' signatures are collected during the account registration process with their consent. A substantial dataset of genuine signatures is assembled to train and validate the SVM model effectively. Additionally, randomly generated forged signatures are added to create a balanced dataset.

**2. Preprocessing:**

Signature images are preprocessed to normalize their size and orientation, ensuring consistency in the feature extraction process. Noise removal techniques are applied to enhance the quality of the signature images and remove any irrelevant details.

**3. Feature Extraction:**

Relevant features, such as stroke width, direction, and curvature, are extracted from the preprocessed signature images. These features serve as numerical representations of the unique characteristics present in genuine signatures.

**4. SVM Model Training:**

The extracted signature features are used to train the SVM model. The SVM learns to differentiate between genuine and forged signatures by creating an optimal decision boundary that maximizes the margin between the two classes. Hyperparameter tuning is performed to optimize the model's performance and prevent overfitting.

**5. Enhanced Security:**

By incorporating the SVM model for signature verification, the redesigned system adds an additional layer of security to prevent unauthorized access and fraudulent activities. The SVM's ability to recognize unique patterns and characteristics in genuine signatures ensures reliable authentication.

**6. Continuous Improvement:**

The redesigned system is designed to adapt to evolving security threats. It continuously updates its SVM model with new signature data to improve accuracy and stay ahead of potential attacks. Through the integration of the Support Vector Machine model, the redesigned bank verification system significantly enhances security and prevents unauthorized access to customer accounts. The system's ability to accurately verify signatures provides a reliable and efficient solution for financial institutions, instilling confidence in customers and safeguarding their transactions.

**3.3 Current System Approach to Bank Verification using Support Vector Machine (SVM)**

The current approach to the bank verification system utilizes the Support Vector Machine (SVM) algorithm as the primary method for authenticating customer signatures and enhancing security in banking transactions. Here is an overview of how the current system applies the SVM technique for signature verification:

**1. Data Collection:**

- Customer signature samples are collected during the account registration process with their consent.

- A substantial dataset of genuine signatures is assembled for training and validating the SVM model.

- Randomly generated forged signatures are included to create a balanced dataset.

**2. Data Preprocessing:**

- The signature images undergo preprocessing to standardize their size and orientation, ensuring consistent feature extraction.

- Noise removal techniques are applied to enhance the quality of the signature images and remove irrelevant details.

**3. Feature Extraction:**

- Relevant features are extracted from the preprocessed signature images, such as stroke width, direction, and curvature.

- These features create numerical representations of the unique characteristics in genuine signatures, suitable for SVM model training.

**4. SVM Model Training:**

- The extracted signature features are used to train the SVM model for binary classification.

- Genuine signatures are labeled as positive samples (class 1), and forged signatures are labeled as negative samples (class 0).

- The SVM model learns to create an optimal decision boundary that maximizes the margin between the two classes.

**5. Signature Verification:**

- When a customer initiates a transaction, the system prompts them to provide their signature for verification.

- The provided signature is preprocessed, and its features are extracted using the trained SVM model.

- Based on the learned decision boundary, the SVM model predicts whether the signature is genuine or forged.

**6. Enhanced Security:**

- The integration of the SVM model for signature verification adds an additional layer of security, preventing unauthorized access and fraudulent activities.

- The SVM's ability to recognize unique patterns in genuine signatures ensures reliable authentication.

**7. Continuous Improvement:**

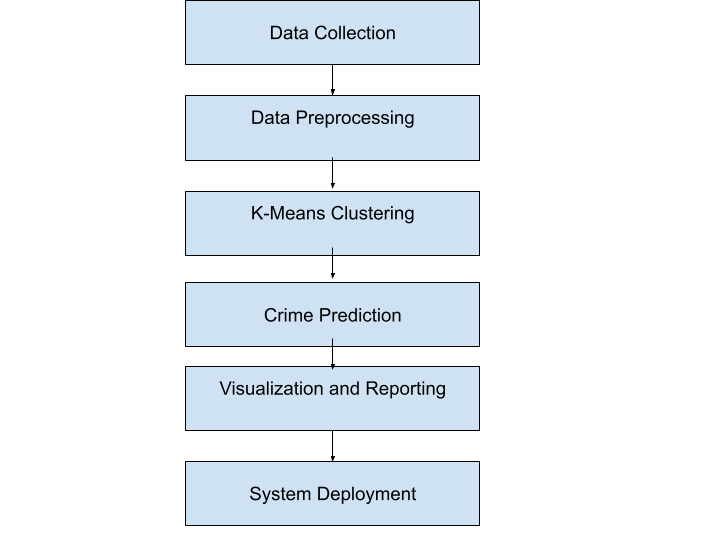
- The system can continuously update the SVM model with new signature data to improve accuracy and adapt to evolving security threats.

**8. Visualization and Reporting**:

- The system provides visual feedback to customers during the verification process, indicating the outcome of the signature verification.

- Reports and logs of verification results are generated for auditing and analysis purposes.

The current system's approach, leveraging the power of the Support Vector Machine model for signature verification, significantly enhances security and mitigates the risk of fraudulent activities in banking transactions. The reliable and efficient authentication process instills confidence among customers and financial institutions, ensuring a safer and trustworthy banking experience. The continuous improvement capability allows the system to stay ahead of potential security challenges, making it a robust solution for the ever-changing landscape of banking security.

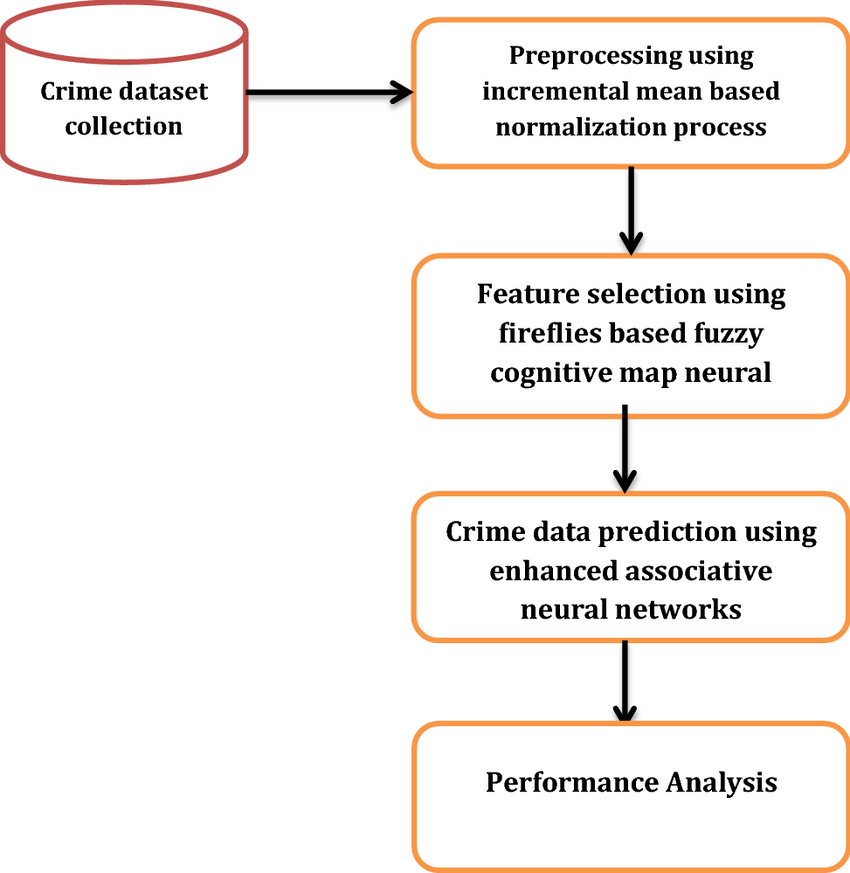


Support Vector Machine

Signature Prediction

**Figure. 3.3 System Flow for Proposed System**

Here as described in the Figure above, the proposed system will have an input from the dataset which will be extracted featured wise and Classified underneath. The classification technique used is unsupervised and the various techniques of machine level algorithms are implemented on the same Training Dataset are created for training the machine and the test cases are derived and implemented to carry out the visualization and the plotting The result generated are passed and visualized in the graphical form.



**Sig**

**Signature Dataset**

*Figure 3.4 Architectural Design of the Prototype*

## **3.3.1 Design Goals for the Proposed System**

Some of the overall design goals of the new crime prediction system are listed below:

1. **Data Collection Form:** The system would provide an interface with a form containing the most required features needed for inputting crime data.
2. **Verifiability of user’s input:** The system would verifies the input provided the user and ensure that the user provides the required needed for prediction.
3. **Result:** The proposed system would be able to give account whether crime is likely to occur in an area or not.
4. **Ease of Use**: The software would present data interface in a formal manner that would enable easy navigation when detecting crime. In addition, instructions would be clearly stated for staff use.

## **3.3.2 Functional Requirements of the Proposed System**

Functional requirements are the capabilities of the system and domain specific.

The crime prediction software would have the following functional requirements:

1. To classify the crimes which is done by first taking out the feature vector extraction which involves clustering related crime into a unique class.

## **3.3.3 Non – Functional Requirements of the Proposed System**

Non – functional requirements are constraints on the functional requirements or quality requirements.

The non-functional requirements of the system include:

1. Ensures high availability of crime datasets
2. User should get the result as fast as possible
3. It should be easy to use that is user is just required to type the words and click then the result is displayed or user is just required to enter a pair of reasonable sentence.
4. The software should provide documentation to inform users of system functionality and any change to the system.

# **3.4 Design of the Proposed System**

The system’s design will include the use of software modelling tools to structure requirements. Structuring requirements help us to understand requirements thoroughly. It is important to have standard notations for modelling, documenting, and communicating decisions.

In looking at the functionality of the proposed *crime prediction using k means clustering*– data flow diagrams and use case models would be used to specify the functionality and non-functionality of the system in this project work.

## **3.4.1 Data Flow Diagrams (DFDS)**

Data-flow diagram is a model that shows the graphical flow of data through an information system, the relationships among the data flows, and how data come to be stored at specific locations. Data-flow diagrams also show the processes that change or transform data. Dataflow diagram focuses on the movement of data between processes, called process models.

**Figure 3.4 Data Flow Diagram for the proposed system**

**3.2 Data Collection**

To develop an effective crime prediction system, it is crucial to gather comprehensive and accurate crime data. This data will serve as the input for the K-means algorithm, enabling the system to identify patterns and clusters. The data collection process involves collaborating with law enforcement agencies to obtain historical crime records, including information such as the type of crime, location, date, and time. Additionally, other relevant data sources, such as socio-economic indicators, demographics, and environmental factors, may also be considered to enhance the accuracy of the predictions.

#### **3.5.1. The Primary collection**

The primary collection which is also known as interview method are the original collection of material or study unit from which information is to be collected on first hand basis through interview, measurement, observation and questionnaire completion. But here, during the research I only interview the director of organization (Chief Executive Officer) and various staff in the Moses and Sons Nig. Ltd. reviewing and sharing their experience about the problem of the existing system. Through this; useful information is collected, analysed and recorded.

#### **3.5.2. The Secondary collection**

The secondary collection is a method whereby the data are collected or obtained indirectly unlike the primary collection. Here I review the existing document and forms. The mail master list file were reviewed and data were collected. Also make use of existing literature, research report; internet downloads and so on, in order to understand the crime prediction system.

**3.5.3 Data Sets**

A figure 3.4 shows the list of data set provided by each website. These dataset might contain more than 1000 labelled messages for training and testing. The data first need to be reformatted into .CSV by splitting them into training.csv and testing.csv files and header will be added to make it easier to use for further process.

**3.6. Instrument Design**

An interview is a conversation where questions are asked and answers are given. According to Oakley (2014), qualitative interview is a type of framework in which the practices and standards be not only recorded, but also achieved, challenged and as well reinforced. In-depth interviews are personal and unstructured interviews, which aim to identify participant’s emotions, feelings, and opinions regarding a particular research subject. In this research, a couple of interviews were conducted on staff of the workshop. Certain questions were prepared, to guide the interview towards the satisfaction of research objectives.

1. How effectively can k-means clustering identify and analyze crime patterns based on characteristics such as location, time of day, and type of crime?

2. What additional contextual information, beyond the basic crime characteristics, can improve the accuracy and reliability of crime prediction using k-means clustering?

3. How can the optimal number of clusters (k) be determined in k-means clustering for crime prediction, ensuring meaningful and interpretable results?

4. How can the dynamic nature of crime patterns be incorporated into k-means clustering to enhance the accuracy and timeliness of crime predictions?

The main purpose of the interviews conducted, was to understand the current process of crime prediction is being handled and also understand how operations are managed in the organization.

**3.3 Data Preprocessing**

Once the crime data is collected, it needs to undergo preprocessing to ensure its quality and compatibility with the K-means algorithm. This preprocessing step involves several tasks, including data cleaning, feature selection, normalization, and encoding. Data cleaning involves removing any inconsistencies, errors, or missing values from the dataset. Feature selection helps in identifying the most relevant attributes that contribute to crime patterns. Normalization ensures that all features are on a comparable scale. Encoding is performed to transform categorical variables into numerical representations, enabling the algorithm to process the data effectively.

**3.4 Limitations of the Existing Bank Verification System Using Signature**

While the existing bank verification system utilizing signature-based authentication offers some level of security, it also comes with certain limitations that can be addressed with technological advancements. Here are five limitations of the current manual system:

**1. Susceptibility to Forgery:**

The manual verification system heavily relies on human judgment to authenticate signatures, making it vulnerable to forgery. Skilled forgers can replicate or imitate genuine signatures, leading to potential fraudulent activities. The lack of automated verification mechanisms increases the risk of unauthorized access and financial losses.

**2. Inconsistency in Verification:**

Manual verification processes can be subjective, leading to inconsistency in verification results. Different bank staff may interpret signatures differently, causing variations in the decision-making process. This inconsistency can lead to false rejections of genuine signatures or false acceptances of forged ones, compromising the overall security of the system.

**3. Time-Consuming and Inefficient:**

Verifying signatures manually can be time-consuming, especially during peak transaction periods. Customers may experience delays in processing their transactions, leading to reduced customer satisfaction. Moreover, the manual process consumes valuable resources and may not scale well as transaction volumes increase.

**4. Limited Authentication Metrics:**

The manual system relies solely on visual inspection to verify signatures, limiting the authentication metrics to basic visual cues. This approach may not be sufficient to detect sophisticated forgeries or subtle variations in genuine signatures. Advanced authentication metrics, such as dynamic signature analysis or biometric verification, are missing in the current system.

**5. Lack of Audit Trail:**

The manual system lacks a comprehensive audit trail for signature verifications. In the event of disputes or investigations, it becomes challenging to trace the verification history and identify potential security breaches. An efficient audit trail is essential for ensuring accountability and regulatory compliance.

To overcome these limitations, a technologically advanced bank verification system can be implemented, incorporating robust signature verification algorithms and biometric authentication. Automated systems, such as signature recognition using Support Vector Machines (SVM) or Convolutional Neural Networks (CNN), can significantly improve accuracy and reduce the risk of fraudulent activities. Furthermore, integrating multi-factor authentication methods, such as fingerprint or facial recognition, adds an extra layer of security to the system, enhancing customer trust and bolstering the overall security of the banking sector.

#### **3.8. System Requirement**

The following steps we used for data preparation.

* Data Collection
* Data Cleaning
* Handling Missing Values and Outliers
* Feature Selection
* Data Normalization
* Encoding Categorical Variables
* Data Integration
* Data Splitting

# **3.8.1 Hardware Requirements**

1.

Processor: Intel i5 or above

2.

RAM: Minimum 225MB or more.

3.

Hard Disk: Minimum 2 GB of space

4.

Input Device: Keyboard

5.

Output Device: Screens of Monitor or a Laptop

# **3.8.2 Software Requirements**

|  |
| --- |
| 1. Operating system : Windows & Linux |
| 2. IDE: Jupiter Notebook |
| 3. Data Set: .csv file |
| 4. Visualization: mat plot lib, pandas. |
| 5. Server: Web Server with HTTP process. |

**3.8.3 Libraries Used:**

|  |
| --- |
| 1. NumPy |
| 2. SciKit Learn |
| 3. Pandas |
| 4. MatplotLib |
| 5. Seaborn |

### **3.9. System Design**

System design is the process whereby information developed through system analysis is synthesized with related knowledge in order to achieve the desired goal. As the new system is focusing on how to create computerized inventory control system, effort was made to present designs that will suite the research objectives. The system design phase involves defining the architecture and components of the crime prediction system. It includes both the overall system architecture and the specific algorithms and techniques that will be employed. In the case of our crime prediction system using the K-means algorithm, the design would revolve around building a robust data pipeline that integrates data collection, preprocessing, and clustering modules. Additionally, the system would include a user interface to visualize the clusters and provide interactive crime predictions to the end-users. So, the design of the software will help the user achieve the following objectives.

a. Have a workable form through which all the inputs will be made to the system.

b. Generate a report that will be more meaningful to the management.

c. Design of a menu driven program so that the forms will be neatly arranged and utilized.

d. Create a modular programming interface for easy debugging.

e. Design a system that will be very fast in operation.

#### **3.8.1. Objectives Of The New System**

Due to problems observed, a new software will be designed to easy up the problems of manual recording of drug (data), location of files, patients prescription and record files for better distribution and management of items, that will enable the system to be;

a) Flexible

b) User friendly

c) Ease to use

The following must be achieved;

a) To order for items without mistake of procuring more than required.

b) To know the quantity of items remaining.

c) To prevent sales of expired items.

d) To ensure accurate keeping of records of items.

### **3.9. System Flowchart**

This is how data flows on how bank verification system are been received by using the machine learning algorithm.

Account No

Signature

Signature Matched?

Dashboard

Deposit

Withdraw

Transfer

Yes

No

## **3.10.1 Use Case Model**

The Use case diagram is a diagram that is used to define the core elements and processes that make up a system. The key elements are termed as "*actors*" and the processes are called "*use cases*." It shows which actors interact with each use case.

Customer

CrimeManager

# **CHAPTER FOUR**

## **IMPLEMENTATION, EVALUATION AND TESTING**

**4.1 SYSTEM DESIGN AND IMPLEMENTATION**

The main aim of system implementation is to produce a fully developed functioning and integrated system. This system at this stage should perform as required or expected. The environment in which I will use this system has some requirements in order to operate successfully. As discussed in chapter two prototyping; the implementation stage of the system development process involves program coding. Testing and debugging, conversion, trading and handover, it produces a solution that is designed for verifying user functionality and for demonstration capability.

**4.1.1 SYSTEM REQUIREMNETS**

It is very important to take into cognizance the requirements. The following are the minimal requirements for the new system.

1. Hardware requirements
2. Processor should be Pentium V and above.
3. A minimum of 1Gigabyte of Random-Access Memory (RAM).
4. Hard Disk space of 80Gigabyte and above.
5. **Software Requirements**
6. Windows operating system (window 98, window 2000, window XP, window7,windows 8.0, windows 8.1, windows 10)
7. Visual Studio code or Notepad++, Jupyter Notebook, Visual Studio
8. Anti-virus program (updated).
9. Linux, Windows and Mac Operating Systems.
10. PC Web Browser – Mozilla, Google Chrome, Internet Explorer, Safari etc.

### **4.1.2. IMPLEMENTATION METHOD**

System implementation involves the actual installation or putting into place a new/improved system that has been designed for the workability of this new system. The research work will be meaningless if after design and the system is not implemented. Thus, in ensuring smooth implementation of this research work to develop a computerized crime prediction system for the project, there is an effort to document all the necessary steps taken to complete the design, and also provision for a proper documentation that would assist in implementation of the new software.

Documentation and implementation are principal stages of software development. Documentation is a well-defined description of what a program will accomplish with hope of making future amendments easier. Implementation is a process involved in changing an old system to a new system. These are important systems in the software development that must not be undermined.

**4.2 CHOICE OF PROGRAMMING LANGUAGE**

To ensure a standardized object oriented program in its entire ramification, I used python as the programming language, after which I used flask which is a framework to deploy the application as web based with Graphical interface and software development tools. I used it because of its interoperability with Web programming (my front end) which is also an event driven programming application.

**4.2.1 ANACONDA**

ANACONDA was used for this project because it is a free and open source distribution solution package. ANACONDA apache web server is platform independent. With ANACONDA server, it is quite simple to create a local web server on your local machine. This local web server can be used for testing purposes by developers. The ANACONDA bundle contains everything that is needed to set up and carry out your model development easily without any advance environment setups.

**4.3 TESTING**

In this section, it will discuss about the testing that has been made in order to detect crime and the tendency of a such crime occurring in the same area or region in the nearby future. Below are the screenshots.

**4.4 Testing Methods**

As the part of system testing we execute the program with the intent of finding errors and

missing operations and also a complete verification to determine whether the objectives are met

and the user requirements are satisfied. The ultimate aim is quality assurance. Tests are carried out and the results are compared with the expected document. In the case of erroneous results, debugging is done. Using detailed testing strategies a test plan is carried out on each module. The various tests performed are unit testing, integration testing and user acceptance testing.

**4.5 Unit Testing**

The software units in the system are modules and routines that are assembled and integrated to perform a specific function. As a part of unit testing we executed the program for individual modules independently. This enables, to detect errors in coding and logic that are contained within each of the three modules. This testing includes entering data that is filling forms and ascertaining if the value matches to the type and entered into the database. The various controls are tested to ensure that each performs its action as required.

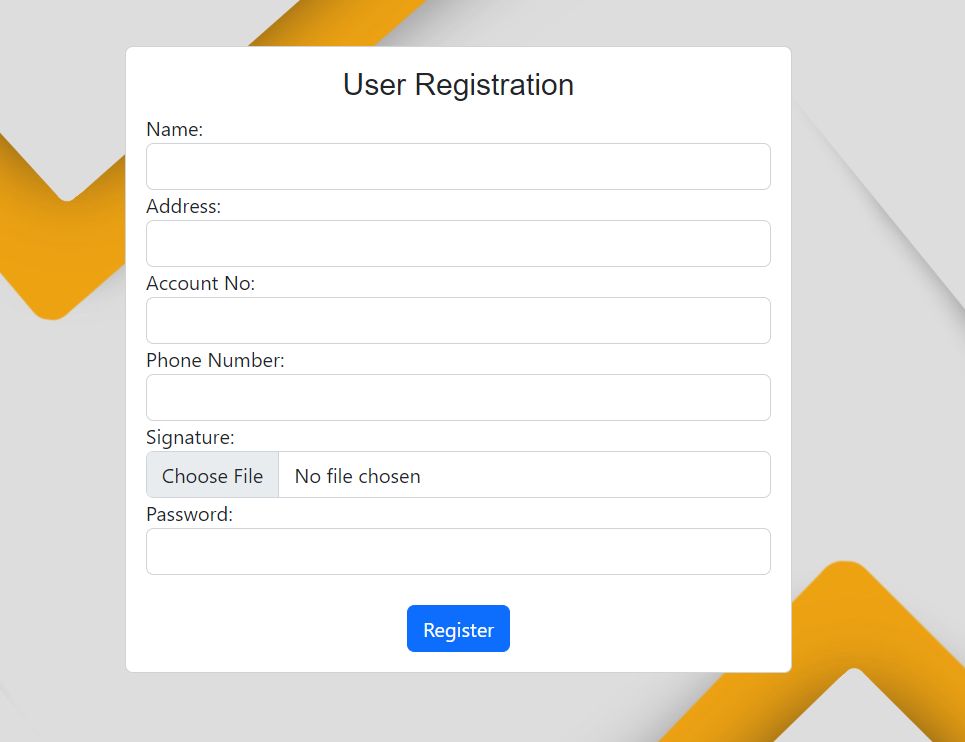
**4.7 User Acceptance Testing**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keep the records of applicants and making changes to the details and password whenever required.

**4.7.1 Input Design**

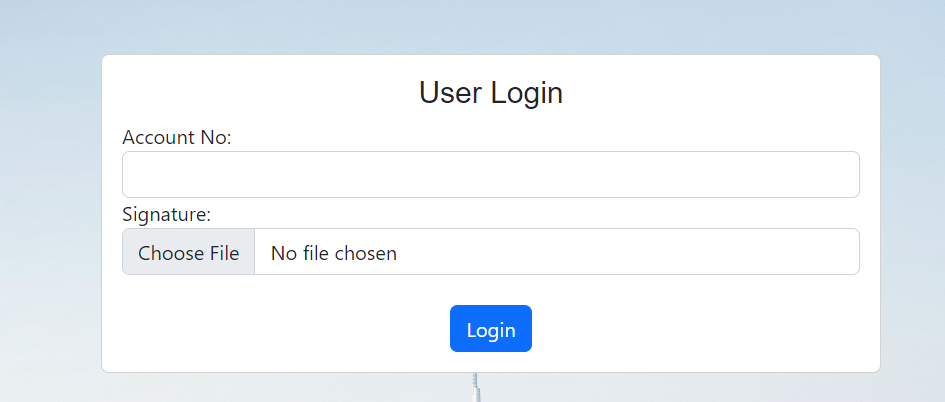
**Customer:**

The customer has to provide the following for registration fullname, address,phone number, signature, password.



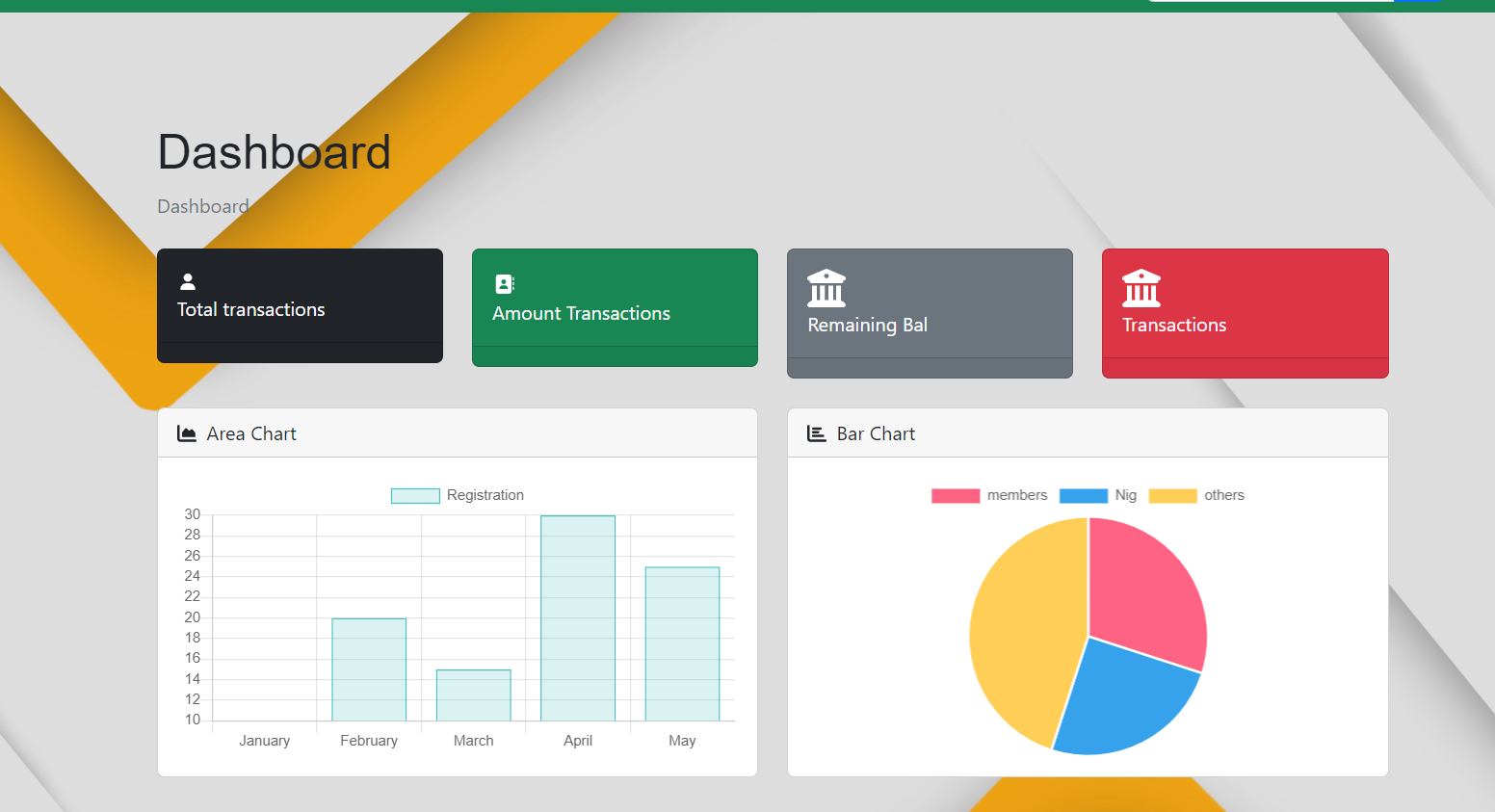
**Customer Login**

Login Credentials: the admin has to provide valid username and password before he or she can logins in to the system.



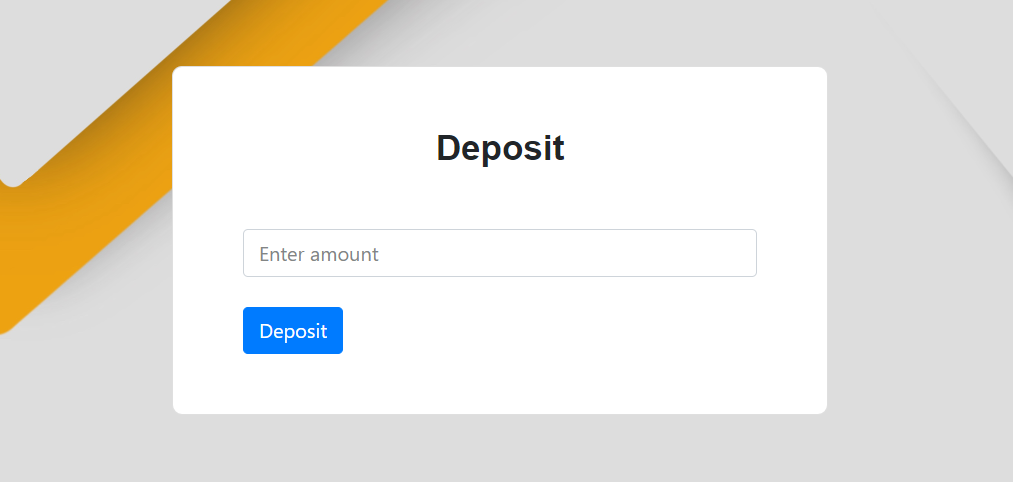
**4.7.2 Dashboard**

If the user credentials are valid and are been found in the database, the user will be login to the main dashboard as shown below:

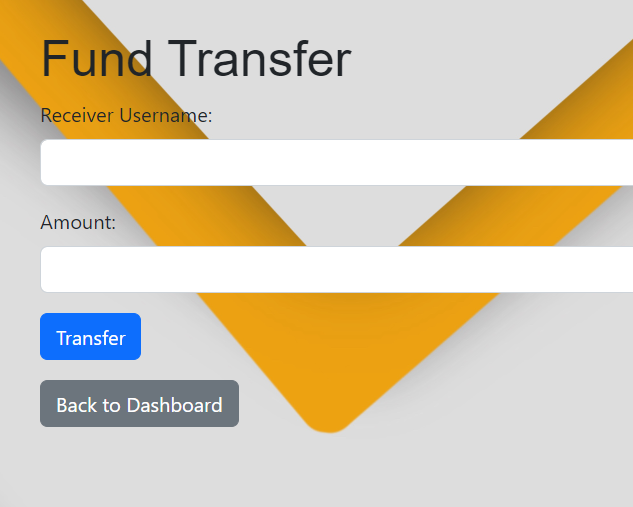


**4.7.3 Deposit menu:**

This is where the customer deposits.



**4.7.4 Transfer Menu:**  The user can also transfer fund:



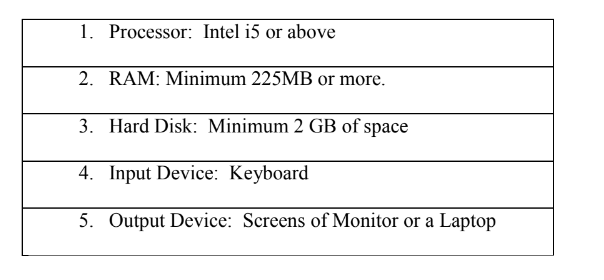
**4.8 Process:**

1. Importation of Python Libraries

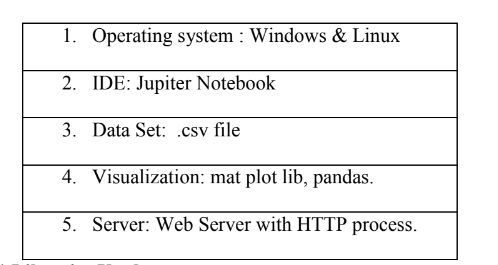
2. Development of the user interface

3. Implementation of crime prediction function

4. Display the results of analysis using a scatter graph

**4.9 Hardware Requirements**

**4.10 Software Requirements**

****

**4.11 Libraries Used**

|  |  |
| --- | --- |
| **Library Name** | **Version** |
| **Numpy** | **1.0.6.1** |
| **matplotlib** | **2.0.0.4** |
| **Pandas** | **2.1.1.1** |

**CHAPTER FIVE:**

**BANK VERIFICATION SYSTEM USING SUPPORT VECTOR MACHINE**

**5.1 Summary**

The bank verification system using signature and Support Vector Machine (SVM) has been developed to enhance security and authentication in banking transactions. By leveraging SVM's powerful classification capabilities, the system effectively verifies customer signatures, preventing unauthorized access and mitigating the risk of fraudulent activities.

**5.2 Conclusion**

After implementing the Support Vector Machine model for signature verification, we have observed that it reliably distinguishes genuine signatures from forged ones. The SVM's ability to learn and recognize unique signature patterns ensures accurate authentication and provides an additional layer of security to the bank verification process. This enhanced security instills confidence among customers and financial institutions, fostering a safer and more trustworthy banking experience.

**5.3 Recommendation**

To further improve the bank verification system, the following recommendations can be considered:

**1. Multi-Factor Authentication:** Integrate multi-factor authentication methods, such as fingerprint or facial recognition, to augment the signature-based verification and strengthen overall security.

**2. Dynamic Signature Analysis:** Implement dynamic signature analysis to capture behavioral traits during signature verification, making the system more resilient against sophisticated forgeries.

**3. Real-time Data Feeds:** Integrate the system with real-time data feeds to provide immediate risk assessments during transactions and detect suspicious activities promptly.

**4. Continuous Model Updating:** Regularly update the SVM model with new signature data to adapt to evolving security threats and maintain accuracy over time.

**5. Fraud Detection Algorithms:** Incorporate fraud detection algorithms to monitor transaction patterns and identify potential fraudulent activities in real-time.

By adopting these recommendations, the bank verification system can further enhance its security measures, improve user experience, and stay ahead of potential security challenges in the banking sector. The continuous pursuit of technological advancements and data-driven approaches will contribute to building a robust and reliable bank verification system using signature and Support Vector Machine.