Secure Encryption of Documents and Smart Complaint Management for Banking Applications

Tirthraj Mahajan[1], Advait Joshi[2]

[1]Tirthraj Mahajan; SCTR's Pune Institute of Computer Technology, Computer Engineering, Pune, Maharashtra, India, tirthraj2004@gmail.com
[2]Advait Joshi; SCTR's Pune Institute of Computer Technology, Computer Engineering, Pune, Maharashtra, India, advaitkjoshi@gmail.com

Abstract

This paper explores the integration of secure authentication and document encryption in web applications, focusing on banking systems. It examines the theoretical foundations and practical implementations of cryptographic techniques such as SHA-256, RSA, and modern encryption algorithms. Additionally, it discusses the challenges and best practices for ensuring data security and privacy. A banking web application case study is presented to illustrate the real-world application of these security measures. This paper also proposes a method to arrange users' complaints in descending order of priority using natural language processing.

Keywords: Securing Authentication, Document Encryption, Cryptographic Algorithms, Token-based authentication, Natural language processing

1. Introduction

In the digital age, banking web applications have become an important tool for managing financial transactions. These applications handle sensitive data and personal information. Thus, ensuring security is paramount in such applications. The increasing prevalence of cyber threats such as data breaches, unauthorized access, identity thefts, etc. requires robust security measures. The paper's primary objective is to analyze and implement secure authentication and encryption methods for secure document uploads, in the context of web banking applications. The focus of the paper is utilizing cryptographic techniques, specifically Scrypt for hashing, RSA for asymmetric encryption, AES for symmetric encryption, and JWT for secure Token-based authentication. By integrating these methods, the paper aims to enhance the security of user authentication processes and the encryption of sensitive documents. This will ensure that user data is protected both in transit and at rest, providing a secure environment for banking operations.

Scope: This paper will concentrate on the following key areas within the domain of secure authentication and document encryption:

1) Cryptographic Hash Functions (Scrypt): An exploration of Scrypt and its properties, and its application in securing passwords
2) Asymmetric Cryptography (RSA): An analysis of the RSA algorithm, including key generation, encryption, and decryption processes
3) Symmetric Cryptography (AES): Discussion of AES encryption techniques for securing documents.
4) Token-Based Authentication (JWT): Examination of JWT, its structure, and its application in secure authentication.
5) Integration of Authentication and Encryption: Strategies for combining secure authentication mechanisms with document encryption to provide a seamless and secure user experience.

Along with all the banking functionalities, a bank application accepts user complaints. For a bank with a large user base, it becomes difficult to scan through all the complaints manually. These complaints can vary in urgency levels,

depending on the severity of the issue. Urgent and sensitive complaints must be resolved with priority. For example, any complaint related to unauthorized transactions through card or Internet banking must be addressed at the earliest to avoid any further mishaps. To make sure that urgent complaints and queries are flagged and addressed on priority, there is a need to automate complaints management in bank applications. Modern techniques of natural language processing and deep learning can be used to develop such a system.

2. Literature Survey

In [1], a Support Vector Machine (SVM) is used to classify newspaper headlines. They have used TF-IDF vectors along with SVM and have claimed good accuracy over a small dataset as compared to other models which need a bigger dataset to reach a good accuracy. Further improvements on SVM-based models are proposed in [2] where they have introduced the concept of uneven margins. The even margin-based model outperforms other models for the task of information extraction. Though there is a difference between information extraction and natural language processing, there is a possibility that this newly proposed model will give better results for NLP tasks too. In [3], SVM is used for sentiment analysis on Japanese text. Like [1], they also have used the TF-IDF. They have compared the accuracy of TF-IDF, CNN and BPCNN. Out of the three, TF-IDF shows the highest accuracy which consistently is distributed around 90% for all the 70 experiments performed. In [4], they compared the results of sentiment analysis on the Twitter dataset. They have used a total of 5 machine learning algorithms and 5 ensemble algorithms. The research reveals that the ensemble algorithms show 3% to 5% better performance as compared to other machine learning algorithms. Similar work is done in [7] on social media posts, blogs, and reviews of products. They have used WordSenses and Synsets to improve classification, at the cost of increasing the size of the feature vector, which demands higher computation power. In [5], they used SVM for the classification of documents in the Hindi language in predefined categories. They have claimed a 100% accuracy. Part of speech tagging is an important technique in NLP. It is mainly used for information retrieval. SVM is used for tagging part of speech in [6] for the Malayalam language. With 20,000 words in a Lexicon, the model shows 63% accuracy, whereas for a Lexicon with 1,80,000 words, the accuracy is 94%. Hence, it can be said that SVM classification performs well in identifying the part of speech. Similarly, in [8], they have used an SVM-based approach to tag parts of speech for the Telugu language. SVM is used for the classification of natural language data available in different forms. SRS (System Requirement Specification) document analysis is performed in [9]. They have also used SVM SVM-based approach for extracting elements in SRS documents. The proposed model has an F-measure of 72.1%.

In [10], along with the commonly used machine learning algorithms, they have tried the deep learning approach as well. They have used artificial neural network (ANN) for NLP. In [11], artificial neural networks and convolutional neural networks are used and compared for the classification of poets. They have used TF-IDF for ANN and word embeddings for CNN. They found the multilayered perceptron (ANN model) has the highest performance. Based on all the related studies, we decided to train two models – one on SVM and the other on ANN.

Cryptographic hash functions are essential in cryptography to achieve security goals such as verifying authenticity, creating digital signatures, implementing digital time stamping, and authenticating entities. They are also closely linked to other key cryptographic tools, such as block cyphers and pseudorandom functions. A cryptographic hash function is an algorithm that transforms a message of any length into a fixed-length hash code or digest. Hash functions are vital in cryptography, serving various security purposes [12]. In [13], the authors discuss the distinction between traditional hashing and key derivation functions. Passwords are never stored in plain-text format; instead, their hashes are generated and stored in the database. SHA256, SHA512, etc. are a traditional way of storing passwords and are not secure any more as they are vulnerable to cracking using methods like lookup and Rainbow tables. This is the reason why, we decided to use one of the latest key derivation functions, Scrypt. It is a hashing algorithm which makes use of a password-based key derivation function and takes a large computation cost. For encryption of documentation, we proceeded with the Advanced Encryption Standard (AES). The current Digital Encryption Standard (DES) is inadequate for today's data security requirements due to its short 56-bit key, which can be easily broken by brute force

attacks according to [14]. [15] explains the process of AES Algorithms. The AES algorithm supports combinations of data and key lengths of 128, 192, and 256 bits. It processes data in 128-bit blocks, which are divided into four basic operational blocks. Full encryption involves Nr rounds of iteration, with Nr being 10, 12, or 14 rounds for key lengths of 128, 192, and 256 bits, respectively. The cypher key is used to encrypt using the public key and decrypt using the private key of the user. For this, we have used the RSA algorithm [16] as it is the most used for such types of asymmetric key encryption. Also, for authentication purposes, we took care of the SQL Injections using prepared statements and regex expression matching. These injection attacks can be used to retrieve information from databases and bypass security mechanisms. These attacks rely on manipulating pre-defined logical expressions within a query by injecting operations that yield always-true or always-false results [17]. In [18], it is explained that JSON Web Tokens offer a scalable solution with notable performance advantages for managing user access in large, decentralized distributed systems. They are better than traditional session-based authentication. We stored information which remains persistent in the JWT. [19] explains the "User update problem" and thus we pass only data that remains persistent as payloads. The payload is then encrypted using an encryption key called the JWT secret. These are called access tokens and are passed along with each subsequent request by the client which is decrypted by the server and the access control data from the token is retrieved.

3. Proposed Methods

This section details the methodologies and procedures used to implement secure authentication and document encryption in our banking web application. We utilized established cryptographic techniques to ensure data security. The methodologies are described under separate subheadings for clarity.

3.1 Input Validation using Regular Expressions

To prevent injection attacks and ensure the validity of user inputs, we used Regex for input validation during registration and login processes.

- **Procedure:**
  1. Name Validation: It has been ensured that the name does not contain any special symbols and is not empty
  2. Username Validation: It has been ensured that the username does not contain any special symbols and is not empty
  3. Email Validation: Ensured that the email complied with the valid format and belonged to a known provider
  4. Password Validation: Ensured that the passwords meet the complexity requirements

Regex validation was the first line of defence, that alone was not sufficient on its own to protect against all types of injection attacks

3.2 User Existence Check Before Registration

To avoid duplicate registrations and ensure the uniqueness of each user, we checked if the user already exists in the database before proceeding with the registration.

- **Procedure:**
  Database Query: After receiving a registration request, a query was performed to check whether the email address was already registered.

  Conditional Registration: If the email address existed, the registration process was halted, and an appropriate message was returned to the user.

Note: To prevent SQL injection, we used parameterized queries or prepared statements for database interactions. This approach ensured that user inputs were treated as data, not as executable code.

3.3 Password Hashing and Salting (Scrypt)

We have used the Scrypt algorithm to hash user passwords. To enhance security, we have also applied salting, a process that involves adding a unique random string to each password before hashing. This prevents attackers from using precomputed hash databases (rainbow tables) to crack passwords.

- **Procedure:**

1. User Registration: When registering a user, a unique salt was generated with a secure random number generator.

2. Hashing: The password and salt were concatenated and then hashed using the Scrypt algorithm.

3. Storage: The resulting hash and salt were stored in the database in a combined format.

- **Implementation:**

```
const { scryptSync, randomBytes } = require('crypto')
class Cryptography{
    generateSalt(){
        return randomBytes(8);
    }
    generateSaltHash(password){
        const salt = this.generateSalt().toString('hex');
        const hashedPassword =  scryptSync(password, salt, 64).toString('hex');
        return `${salt}:${hashedPassword}`;
    }
}
```

3.4 Asymmetric Key Encryption using RSA

RSA was employed to encrypt the AES keys used for document encryption, ensuring that only authorized users with the correct private RSA key can decrypt the documents

- **Procedure:**

1. **Key Generation**: RSA key pairs (public and private keys) were generated for each user

2. **Key Encryption**: The user's private key was encrypted with the server's secret key, using the user's salt value used as the Initialization Vector (IV). This ensured that the private keys remained secure even if the database was compromised, and could only be decrypted by the server.

3. **Storage**: The public key and the encrypted private key were stored in the database.

By following this procedure, we ensured a robust and secure method for handling key management in our application

- **Implementation:**

```
const { generateKeyPairSync } = require('crypto')
class Cryptography{
    generateKeyPair(){
        let  { privateKey , publicKey } = generateKeyPairSync('rsa', {
            modulusLength: 2048,
            publicKeyEncoding:{
                type: 'spki',
                format: 'pem',
            },
            privateKeyEncoding:{
                type: 'pkcs8',
                format: 'pem',
            }
        })
        return {privateKey: privateKey, publicKey: publicKey}
    }
}
```

3.5 Token Based Authentication

We have implemented JWT (JSON Web Token) for secure token-based authentication. JWTs are used to verify the identity of users after login and ensure secure sessions.

- **Procedure**
  1. **Token Generation:** A JWT was generated after a successful login. The payload contained user information and was signed with the server's secret key.
  2. **Token Verification:** For subsequent requests, the JWT was sent in the HTTP headers and verified on the server with the same secret key.

- **Implementation**

```
const jwt = require('jsonwebtoken')
class JSON_WEB_TOKEN{

    // Init payload of the jwt token

    createPayload(userId,name,username,email,iv,role){
        return {
            uid: userId,
            name: name,
            username: username,
            email: email,
            iv: iv,
            role: role
        }
    }

    createToken(payload){
        // return the signed jwt token
        return jwt.sign(payload, SECRET_KEY, { expiresIn: '7d' })
    }

    validateUserToken(userToken){
        try {
            const decodedToken = jwt.verify(userToken, SECRET_KEY);     // fetch
the decoded token using the secrete key and userToken
            // console.log(decodedToken);

            // If token is not verified, then it will throw an error

            if (decodedToken.exp < Math.floor(Date.now() / 1000)) {     // check
if the token has expired
                return { valid: false, reason: 'Token has expired' };
            }

            // return valid token

            return { valid: true, decodedToken: decodedToken };
        } catch (error) {
            return { valid: false, reason: 'Invalid token: ' + error.message };
        }
    }
}
```

3.6 Symmetric Key Encryption using AES

AES (Advanced Encryption Standard) was used for encrypting documents due to its efficiency and security. AES operates using a symmetric key, which means that the same key is used for both encryption and decryption.

- **Procedure**
  1. **Encryption:** The content of the uploaded document was encrypted with a randomly generated AES key.
  2. **Key Management:** Then the AES key was encrypted using RSA and stored securely.
  3. **Storage:** The encrypted document and the encrypted AES key were stored in the database.

- **Implementation**

```
const { createCipheriv, createDecipheriv } = require('crypto');
function encipherFile(fileContent,salt){
    const cipher_key = randomBytes(32).toString('hex')
    const cipher = createCipheriv('aes-256-cbc', Buffer.from(cipher_key, 'hex'),
salt);
    let encipheredFileContent = cipher.update(fileContent);
    encipheredFileContent = Buffer.concat([encipheredFileContent,
cipher.final()]);
    return {content: encipheredFileContent, key:cipher_key}
}
```

3.7 Integration of Authentication and Encryption

To provide a seamless and secure user experience, we integrated the above authentication and encryption mechanisms.

- **Workflow**
  1. **User Login:** The user logs in and receives the JWT
  2. **File Upload:** The user uploads the document, which is then encrypted using AES. The AES key is encrypted using the user's RSA public key
  3. **File Access:** When the user wants to access the document, the AES is decrypted using the RSA private key and then the document is decrypted using the AES key

Figure 1 explains in detail, the flow of the encryption process.

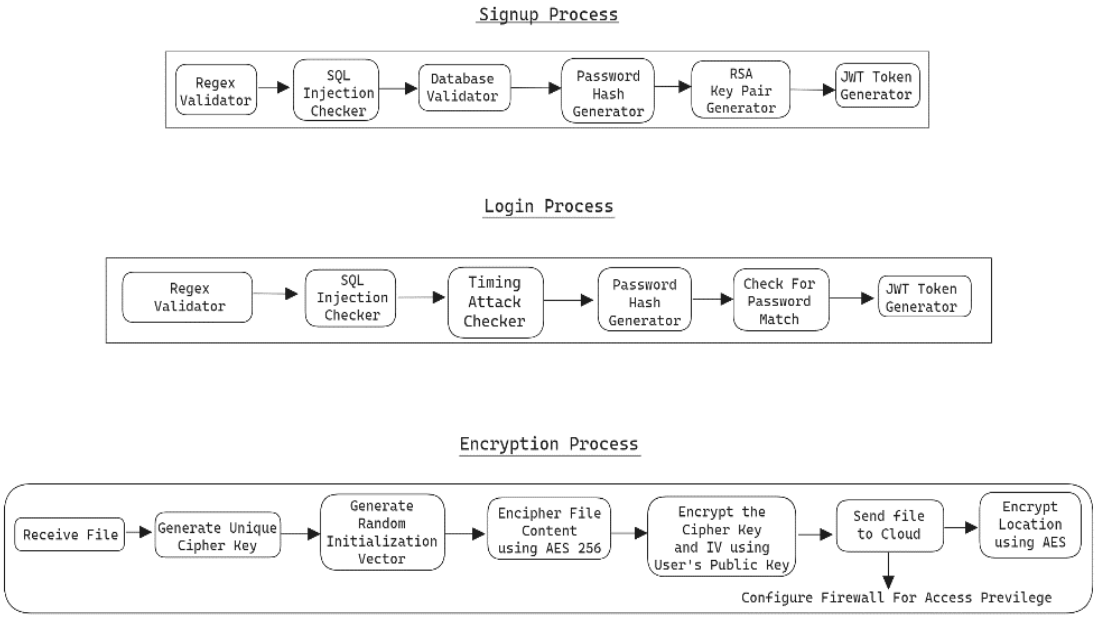

Fig. 1 Encryption flowchart

3.8 Complaint Management Model

To automatically determine the urgency of the complaint, we followed the following steps:

1. Dataset: Created a custom dataset with 222 complaints and corresponding urgency values from 0 to 1, where 0 is the least urgent and 1 is the most urgent.
2. Vectorization: Vectorized the dataset by using tf-idf vectorizer
3. Vocabulary generation: Generated a vocabulary of 600 words from the dataset. Figure 2 shows the word cloud of the vocabulary:

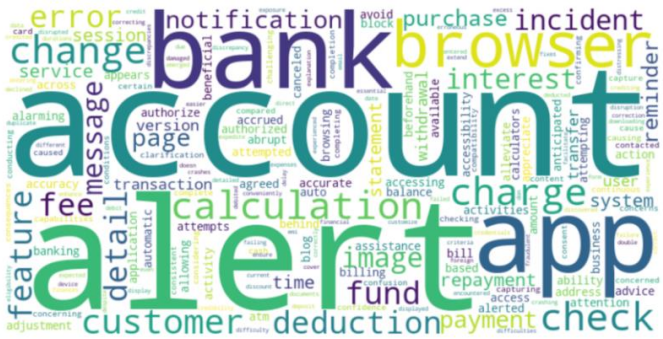

Fig. 2: Word cloud of the vocabulary

4. Splitting data: We kept 80% data for training and 20% data for testing
5. Model training
6. Get the mean squared error from the testing data
7. Input all the complaint sentences to the model and get the complaints in descending order of urgency as the output.

4. Results and Discussion

We trained two models for urgency detection:

1. Support Vector Machine (SVM)
2. Artificial Neural Network (ANN)

Figure 3 shows the summary of the ANN model.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               76928

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 64)                4160

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 89409 (349.25 KB)
Trainable params: 89409 (349.25 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig 3: Model summary

Table 1 shows the comparison of mean squared error, mean absolute percentage error and the R2 score of both models.

Table 1: Comparison of models

Metric	SVM Model	ANN Model
Mean Squared Error (MSE)	0.0097	0.0075
Mean Absolute Percentage Error (MAPE)	0.1591	0.1149
R2 Score	0.8085	0.8514

The histogram in Fig 4 compares the MAPE and the R2 Score of SVM and ANN models. ANN model shows lower error and greater R2 Score than the SVM model. Hence, the ANN model has the best performance. Therefore, we selected this model for our application.

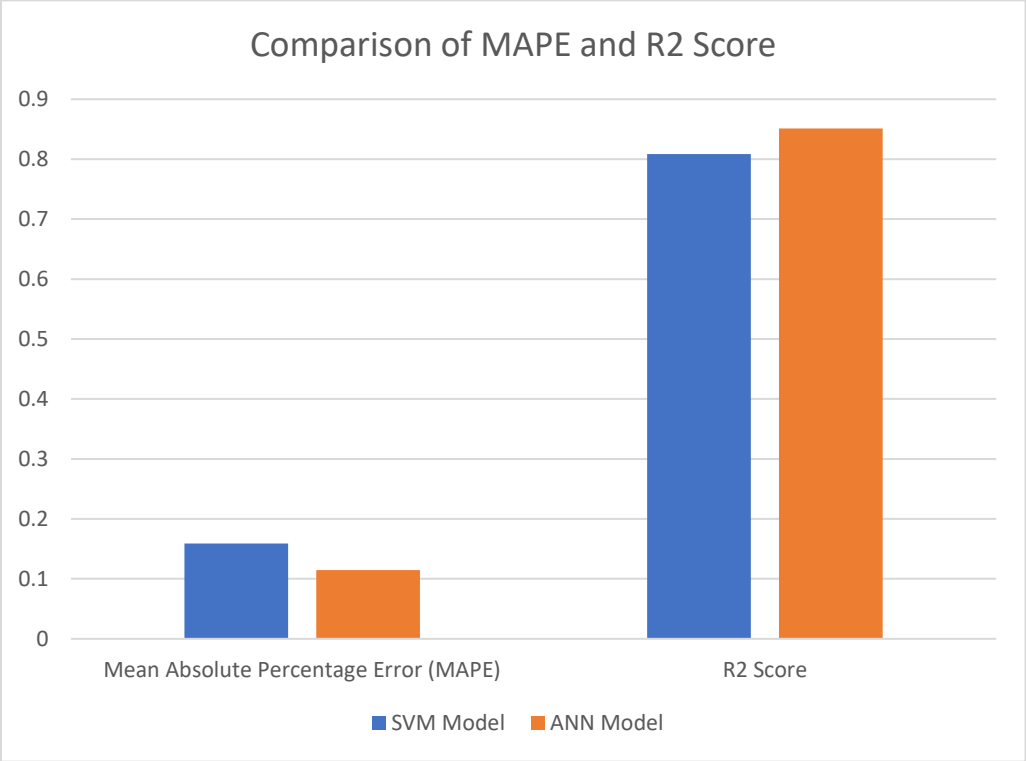

Fig 4: Comparison of MAPE and R2 Score

Example of 5 sample complaints:

1. "I have noticed an unauthorized transaction on my account dated May 10th for $50. This transaction was not made by me, and I am very concerned about the security of my account. I need this resolved immediately as it is affecting my balance and causing me significant stress. Additionally, I would like to know what measures are being taken to prevent such incidents in the future."
2. "My credit card payment due date was suddenly changed without any notification. This caused me to miss the payment, which I believe is unfair and needs to be rectified. I have always been prompt with my payments, and this sudden change has disrupted my financial planning. Please address this issue regarding the expiry of my card, as it was not my fault."

3. "I have been charged twice for the same purchase at your store on April 25th. This has resulted in an unnecessary deduction from my account, and I am now short on funds for other expenses. Please refund the duplicate charge as soon as possible and confirm that this will not happen again. I also suggest reviewing your payment processing system to avoid such errors."
4. "I received my monthly statement, and there is a misprint in my name. It is spelt incorrectly and should be corrected in your records. This kind of error can lead to issues with identity verification and other banking services. Please update your records and send me a corrected statement. I would appreciate confirmation once this is done."
5. "There is a typo in my passbook for the entry on March 3rd. It shows the wrong transaction description, which is a minor issue but needs correction for accurate record-keeping. Maintaining correct records is important for my financial tracking. Please update the passbook entry and send me a corrected version. This will help me keep my financial documents in order."

Table 2 shows the score that the model assigned to each complaint.

Table 2: Urgency score of complaints

Complaint ID	Urgency Score
1	0.8539581
2	0.8328156
3	0.8988212
4	0.67656535
5	0.7136923

Hence, the model will return the complaints in order of IDs: 3, 1, 2, 5, and 4, where 3 is the most urgent and 4 is the least urgent. Fig 5 shows the line graph of the urgency score of all 5 complaints after sorting.

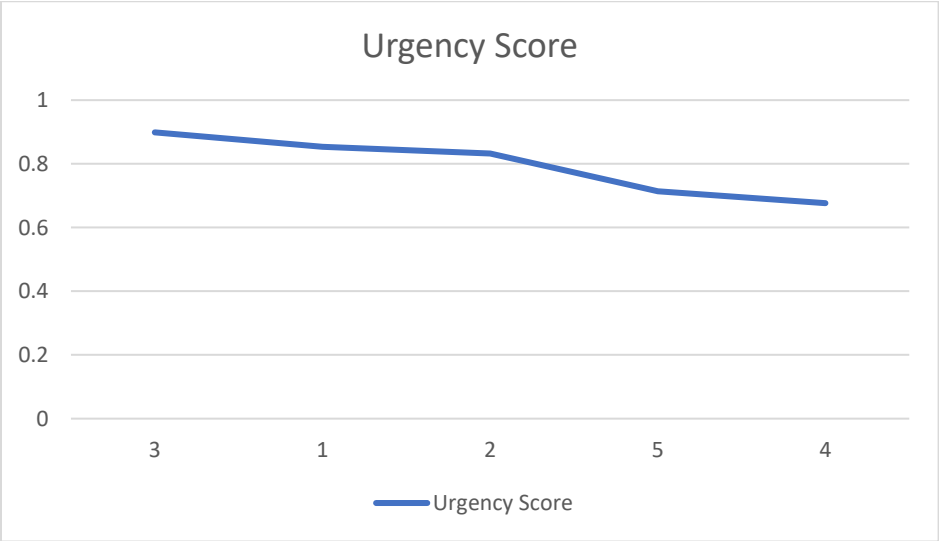

Fig 5: Comparison of MAPE and R2 Score

5. Conclusion

We presented a comprehensive approach to secure authentication and encryption of user documents in a web application. Our solution integrates both symmetric and asymmetric encryption techniques to ensure robust data protection and user privacy. Our approach follows industry best practices, incorporating strong cryptographic methods and secure key management techniques. Future work may include exploring more advanced key management techniques, integrating multi-factor authentication, and continuously updating cryptographic practices to stay ahead of emerging threats. Natural language processing can be effectively used to detect the urgency level of a statement. Regression models based on support vector machine, and artificial neural networks are effective in determining the urgency score of the complaints in the bank app. Using this score, it is possible to sort the complaints, thus adding to faster addressing of highly urgent complaints. The ANN-based model with an R2 score of 0.8514 performs better than the SVM model which has an R2 score of 0.8085. This can be further enhanced by increasing the dataset size and applying LSTMs and transformers. Further research can be done for the classification of complaints into various categories so that they can be transferred to the required department.

References

[1] Rameshbhai, Chaudhary Jashubhai, and Joy Paulose. "Opinion mining on newspaper headlines using SVM and NLP." *International journal of electrical and computer engineering (IJECE)* 9.3 (2019): 2152-2163.

[2] Li, Yaoyong, Kalina Bontcheva, and Hamish Cunningham. "Adapting SVM for natural language learning: A case study involving information extraction." *Natural Language Engineering* 15.2 (2009): 241-271.

[3] Song, Gang. "Sentiment analysis of Japanese text and vocabulary learning based on natural language processing and SVM." *Journal of Ambient Intelligence and Humanized Computing* (2021): 1-12.

[4] Kanakaraj, Monisha, and Ram Mohana Reddy Guddeti. "NLP based sentiment analysis on Twitter data using ensemble classifiers." *2015 3Rd international conference on signal processing, communication and networking (ICSCN)*. IEEE, 2015.

[5] Puri, Shalini, and Satya Prakash Singh. "An efficient Hindi text classification model using SVM." *Computing and Network Sustainability: Proceedings of IRSCNS 2018*. Springer Singapore, 2019.

[6] Antony, P. J., Santhanu P. Mohan, and K. P. Soman. "SVM based part of speech tagger for Malayalam." *2010 international conference on recent trends in information, telecommunication and computing*. IEEE, 2010.

[7] Sharma, Dipti, and Munish Sabharwal. "Sentiment analysis for social media using SVM classifier of machine learning." *Int J Innov Technol Exploring Eng (IJITEE)* 8.9 (2019): 39-47.

[8] Binulal, G. Sindhiya, P. Anand Goud, and K. P. Soman. "A SVM based approach to Telugu parts of speech tagging using SVMTool." *International Journal of Recent Trends in Engineering* 1.2 (2009): 183.

[9] Imam, Ayad Tareq, Aysh Alhroob, and Wael Alzyadat. "SVM machine learning classifier to automate the extraction of SRS elements." *International Journal of Advanced Computer Science and Applications (IJACSA)* (2021).

[10] Naithani, Kanchan, and Yadav Prasad Raiwani. "Realization of natural language processing and machine learning approaches for text-based sentiment analysis." *Expert Systems* 40.5 (2023): e13114.

[11] Ekinci, Ekin, Hidayet Takcı, and Sultan Alagöz. "Poet classification using ann and dnn." *Electronic Letters on Science and Engineering* 18.1 (2022): 10-20.

[12] Gauravaram, Praveen 2007 Cryptographic Hash Functions: Cryptanalysis Design and Application. Ph.D. thesis, Information Security Institute, Faculty of Information Technology, Queensland University of Technology.

[13] Ertaul, Levent, Manpreet Kaur, and Venkata Arun Kumar R. Gudise. "Implementation and performance analysis of pbkdf2, bcrypt, scrypt algorithms." Proceedings of the international conference on wireless networks (ICWN). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016.

[14] A Performance Study on AES algorithms

Prasad, B. D. C. N., and PESN Krishna Prasad. "A Performance Study on AES algorithms." International Journal of computer science and information security 8.6 (2010): 128-132.

[15] A. K. Mandal, C. Parakash and A. Tiwari, "Performance evaluation of cryptographic algorithms: DES and AES," 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 2012, pp. 1-5, doi: 10.1109/SCEECS.2012.6184991. keywords: {Encryption;Memory management;Algorithm design and analysis;Software algorithms;Classification algorithms;Advanced Encryption Standard (AES);Avalanche Effect;Cipher text;Data encryption standard (DES);Secret key},

[16] Xin Zhou and Xiaofei Tang, "Research and implementation of RSA algorithm for encryption and decryption," Proceedings of 2011 6th International Forum on Strategic Technology, Harbin, Heilongjiang, 2011, pp. 1118-1121, doi: 10.1109/IFOST.2011.6021216.
keywords: {Cryptography;Algorithm design and analysis;RSA algorithm;encryption;decryption},

[17] Boyd, S.W., Keromytis, A.D. (2004). SQLrand: Preventing SQL Injection Attacks. In: Jakobsson, M., Yung, M., Zhou, J. (eds) Applied Cryptography and Network Security. ACNS 2004. Lecture Notes in Computer Science, vol 3089. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24852-1_21

[18] Ahmed, Salman, and Qamar Mahmood. "An authentication based scheme for applications using JSON web token." 2019 22nd international multitopic conference (INMIC). IEEE, 2019.

[19] Jánoky LV, Levendovszky J, Ekler P. An analysis on the revoking mechanisms for JSON Web Tokens. International Journal of Distributed Sensor Networks. 2018;14(9). doi:10.1177/1550147718801535, ks. 2018;14(9). doi:10.1177/1550147718801535