Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19020-2

Telecommunication fraud resilient framework for efficient and accurate detection of SMS phishing using artificial intelligence techniques

Devendra Sambhaji Hapase[1] · **Lalit Vasantrao Patil**[1]

Received: 28 October 2023 / Revised: 23 January 2024 / Accepted: 20 March 2024

Abstract
One of the telecommunications' most popular forms of fraud is the short message service (SMS). Mobile users have a valid fear about SMS spam, which disturbs telecoms network operators since it impacts their clients and costs them money. For that, the existing research utilized an artificial intelligence approach to detect SMS phishing in telecommunication. Since SMS text data is unstructured and contains complicated, nonlinear relationships, this process could be difficult. Therefore, this research developed a Fraud Resilient Framework using Enhanced CNN-based SMS Phishing detection. Telecommunication fraud-related datasets are collected. Firstly, the data are preprocessed and cleaned using stemming, tokenization, and the TF-IDF approach. Moreover, to extract the features, the existing research utilized the information gain technique, which is time-consuming. So to overcome these flaws, this research introduces Assimilated Pearson Correlation Coefficient Principal Component Analysis (PCC-PCA) for feature extraction. This research introduces an enhanced Convolutional Neural Network (Enhanced CNN) in which, overcome the exploding gradients, this research introduces Parameterized ReLU which minimizes architecture complexity, regularizing, and early stopping. Then, the retrieved features are used in Enhanced CNN to categorize the ham and spam in the telecommunication network. As a result, when matched to cutting-edge techniques, this proposed solution offers great accuracy and efficiency.

Keywords Deep learning · SMS Phishing detection · Pre-processing · Improved Pearson Correlation Coefficient Principal Component Analysis (PCC-PCA) · Fraud Classification

✉ Devendra Sambhaji Hapase
devenhapase17@gmail.com

Lalit Vasantrao Patil
lalitvpatil@gmail.com

[1] Department of Computer Engineering, Smt.Kashibai Navale College of Engineering, Vadgaon(BK), Pune 411041, India

Published online: 28 March 2024

1 Introduction

Fraud is the most significant source of revenue loss in the telecom business. Every year, tens of billions of dollars are lost due to telecommunications fraud worldwide. In the telecom industry, fraud has always been a problem. Fraudsters have devised several methods for obtaining services for free or for the interests of others. They utilize various methods, including manipulating services, stealing IDs, credentials, or hardware, compromising physical core network security, or replicating equipment [1]. Anomaly patterns vary depending on the scenario and must be discovered and updated as fraudsters refine their methods. The authors [2] emphasise the importance of making the distinction among fraud finding and avoidance. The term "fraud prevention" describes actions made to stop fraud before it even starts. Fraud detection, on the other hand, entails detecting fraud as soon as possible after it has occurred. Subscription fraud, PBX infrastructure manipulation or dial-through scams, complimentary phone scams, expensive rate communication scams, handset thievery, and roaming scams are the six possible instances of fraud that the authors [3] identified.

The necessity to address concerns including fraudulent activity, financial fraud, computer intrusion, and telecommunication service forgery is evident [4]. Because there is an enormous amount of information to analyse and, at same period, only a low amount of illegal call instances that might be consumed as learning data for learning-based systems, fraud prevention in communications appears to be one of the most challenging of these [5, 6]. As a result, this difficulty effectively prevents and restricts learning-based approaches, such as neural-network-based classifiers. In general, rule-based and user-profile-based fraud prevention methods are separated [7]. The second tactic is regarded to be more successful and has become more common in practical uses.

The most popular fraud detection methods at the moment are hard rule-based or machine learning algorithms, which generally use publicly available credit agency information or the credit reports of the users as inputs to identify fraudulent conduct [8]. Because scammers' methods change as they become more familiar with the trends in the data that the ML model or the rules-based approach associates with fraudulent conduct, systems that use conventional information may be less efficient in detecting fraudulent.

Fraudulent credit card transactions have also been detected using deep learning algorithms. The authors propose interleaved sequence RNNs for detecting fraudulent payments, claiming that they achieve gains over a Light GBM baseline design with recall enhancements of at least 3.2 percent in two independent time split databases. In addition, similar deep learning algorithms take been modified for graphs to increase fraud detection [9]. For the identification of financial fraud, the author planned a Semi-Supervised Graph Attentive Network, which outperformed other compared models [10]. Also [11] introduced a hybrid deep learning system that automatically detects phishing SMS after extracting significant elements from SMS messages. It has done better than numerous different standalone artificial intelligence systems because it combines the power of numerous models into one hybrid framework. The suggested architecture for detecting phishing is a powerful hybrid of a pre-trained transformer model.

The author [12] developed a model for detecting smishing that consists of the domain checking phase and the SMS classification phase. To effectively identify SMS phishing, the authors have checked the validity of the URL in the SMS. The authors algorithm meticulously evaluates the authenticity of the URL during the province checking stage.

The SMS categorization stage reads through the message text and extracts certain useful information. A machine-learning method is used to analyse four rank correlation techniques to determine the preeminent feature set for recognising scam communications [13]. The results of the experiment demonstrate that the AdaBoost cataloguing presented more performance. Moreover, the Phisher Cop anti-phishing tool is developed by [14]. Stochastic Slope Succession and Support Direction categorizers, that are the foundation of PhisherCop, demonstrated a median accurateness of 96%, outperforming six other widely used classifiers, such as Naïve bayes, Boosting classifier, Regression, K value based nearaset neighbours selection, tree based approach, it attains a high accuracy. Additionally, the perpetrators of such attacks employ a variety of instruments and techniques [15–17]. Phishing is a tactic that is most frequently used in the modern digital age. Phishing is a dishonest challenge to trick consumers into giving over critical information. These cyberattacks could be used to harm either particular people or significant organisations, based on the attacker's desired objective. Criminals may take use of this technology by delivering malicious email to their targets in order to fool them into disclosing personal information or downloading dangerous software to their machines [18]. By investigating the techniques/strategies that dishonest transmitters and sincere receivers use to conceal/identify SMS-based phishing, this research discusses the lack of understanding of SMS-based phishing [19]. The subsequent are the crucial offerings of this research:

- To extract the features from the call records, the existing research uses an information gain technique but has a high computational time limitation. To overcome the limitations in the existing works, this research introduces improved Pearson Correlation Coefficient Principal Component Analysis (PCC-PCA) for feature extraction.
- Also, SVM is utilized to classify SMS phishing fraud. However, it does not provide high accuracy. So this research proposed an enhanced Convolutional Neural Network (Enhanced CNN) to classify spam or ham.

The organization of the essay is structured as trails: The associated efforts are listed in Section 2. The proposed approach for detecting Enhanced CNN-based SMS phishing has been discussed in Section 3, the findings and discussion have been explained in Section 4, and the inference has been offered in Section 5.

2 Literature survey

In current customary artificial intelligence-based approaches applying for detecting Phishing SMS. This section contains a survey on telecommunication fraud detection challenges,

For assessing the possibility of fraud for each large transfer so that the financial institution may take the necessary precautions to prevent possible criminals from stealing property if the likelihood exceeds a threshold, Zheng et al. [20] introduced a new procreative accusatorial system based method. In order to distinguish among true and false tests in the distribution of the data precisely, the inferential system uses adversarial preparation. This creates a minimax game between a determiner and a power source. It also makes using a deep

denoising auto encoder to good education the intricate probability—based relationship between input features. But because of non-convergence, design variables bounce, become unstable, and never combine. Mode collapse: When the generator failures, just a few sample variations are produced. Reduced gradient: When the differentiator is too effective, the generating gradient disappears and the learner is left in the dark.

Kashmir et al. [21] proposed machine learning techniques (MLTs) to distinguish between legitimate and fraudulent subscribers (SIM Box). The author is motivated by organization techniques used in machine learning and applied in several sciences and engineering disciplines, such as image processing, language detection, and spam email recognition. To identify the relevant qualities, the authors used call records to detect the scam in the input 25 for each client. Using Neural Network (NN) and Support Vector Machine, these attributes are utilized to classify normal and fraudulent subscribers (SVM). However, the SVM method fails when used to large quantities of documents. SVM performs poorly when the data set has additional noise, such as overlapping targeted communications. If there are more characteristics per piece of data than there are training sequence data, the SVM will perform poorly.

Boukari et al. [22] proposed a Naive Bayes algorithm for Smishing and Vishing attacks. Victims of phishing and Smishing scams endure not only financial losses but also social and psychological consequences. The telecoms industry has long lacked a straightforward, workable alternative that safeguards sufferers. The output of a machine learning-based protection system that finds and warns clients about Smishing frauds is summarised in this demo. This method is also used to detect phishing and vishing attacks. The "zero-frequency problem" affects native Bayes networks when a technique gives null value to a categorical data whose subcategory is present in the sample set of data but not in the training sample. A softening strategy should be used to solve this issue.

For the purpose of identifying SMS spam messages, Hameed et al. [23] introduced a novel approach based on numerical particle swarm optimization and fuzzy rule choosing. The author starts by identifying the most important features in the SMS spam dataset. Then, based on the features that were gathered, a collection of fuzzification was created. The outcome for choosing more potent fuzzy rules that reduce model problems and increase performance is a binary particle swarm. The benchmark dataset for SMS spam is used in the experiment. The optimization technique has the disadvantages that the agile approach has a sluggish rate of convergence and that it is simple to fall into a locally optimal in a high-dimensional environment.

To improves the precision of SMS action recognition, Wu et al. [24] planned a new process created on feature improvement and oversampling technology. The three types of features offered are token characteristics, subject structures, and Linguistic Inquiries and Word Count (LIWC). The Adaptive Synthetic Sampling Approach, one of the known oversampling approaches, is used in this paper because of its high performance. The binary optimization technique is then employed to evaluate the 3 different feature categories and select the perfect solution. Finally, the Random Forest classification technique produces the detection results. On complex operations, gradient-boosted trees frequently low performance them in classification accuracy.

Support Vector Machine (SVM) method usage was suggested by Sjarif et al. [25] for SMS spam classificationThe proposed technique was tested with a

publicly available UCI machine learning collection dataset. The outcomes are related to those of three other data mining techniques: Nave Bayes, Multinominal Nave Bayes, and KNearest Neighbor, with $K = 1,3$ and 5, respectively. SVM performs poorly when the set of data has additional sound, such as overlapping targeted communications.

Srinivasarao et al. [29]., suggested a hybrid model based on sentiment analysis and SMS spam categorization. The attributes are extracted using Word2vec data enhancement after the datasets have undergone pre-processing. Subsequently, the characteristics are supplied through equilibrium optimization (EO) and six different feature selection techniques. To categorize SMS messages, the best parts are then fed into a hybrid K-Nearest Neighbors (KNN) and support vector machine (SVM) classifier. Moreover, the optimization algorithm Rat Swarm Optimization (RSO) is applied to enhance accuracy and optimize the network's parameters.

By concentrating on the word semantics, Oswald et al. [30]., created an intention-based method for SMS spam filtering that effectively handles dynamic keywords. Based on thirteen pre-defined intention labels, we extract the short-text messages' textual and semantic characteristics. Furthermore, a number of previously learned NLP (Natural Language Processing) models are used to create the contextual embeddings of the texts. In order to filter as spam or ham, a number of supervised learning classifiers are used after intention scores for the pre-defined labels have been calculated.

As a result, to effectively detect fraud in telecommunication, a novel technique is needed. To overcome the above limitations, this research proposes an effective telecommunication fraud detection network discussed in the forthcoming section.

3 Enhanced CNN-based telecommunication fraud resilient framework

SMS is one of the top extensively utilized forms of communication in the telecommunications industry. Filtering spam messages, detecting and classifying spam occurrences is a challenging task. This research introduces a novel Enhanced CNN-based SMS Phishing detection. The flow diagram of Enhanced CNN-based SMS Phishing detection is shown in Fig. 1. It is classified into three sections such as data preprocessing, feature extraction, and classification. The data are preprocessed and cleansed by the following methods such as tokenization, TF-IDF, and stemming. In addition, this research introduces improved Pearson Correlation Coefficient Principal Component Analysis (PCC-PCA) for feature extraction, which has the advantage of low dimensional data into high dimensional data without loss in information. Then, extracted features are fed into the classification process, in which the existing research is classified as SMS phishing fraud using SVM. To overcome the above-stated limitations and accurately classify the SMS phishing fraud in a telecommunication network, this research introduces an enhanced Convolutional Neural Network (Enhanced CNN) to overcome the exploding gradients. This research introduces Parameterized ReLU, which minimizes architecture complexity, regularizing, and early stopping. As a result, this proposed method achieves high accuracy and efficiency and reduces time compared to the existing techniques. The architecture of the proposed approach is depicted in Fig. 1.

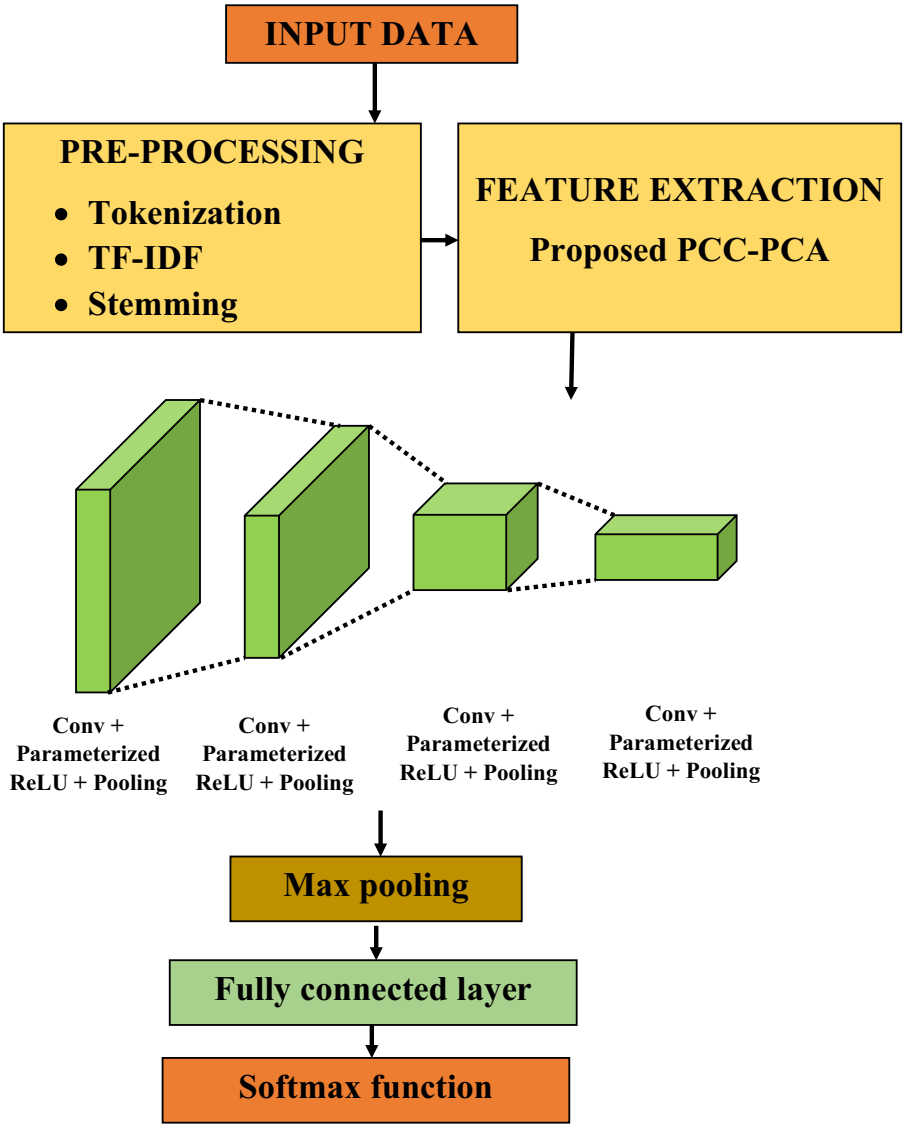

Fig. 1 Enhanced CNN-based SMS Phishing detection

3.1 Preprocessing

Telecommunication fraud-related datasets are collected. To reduce noise and improves a good quality preprocessing method is used. Then, the data are preprocessed and cleansed by the following methods: tokenization, TF-IDF, and stemming. Figure 2 preprocessing methods.

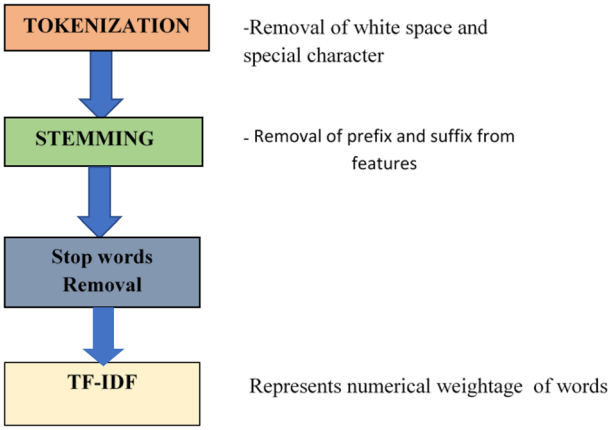

Fig. 2 Preprocessing methods

Tokenization Tokenization is substituting sensitive data with distinctive identifying symbols that preserve all of the material's key details while maintaining security. Textual units are created by preprocessing raw texts. The information must be handled in the three procedures: converting the file to word counts, which is an equivalent first action Words to bag (BOW). The another process, or this phase, removes empty sequences, including scrubbing and filtering. Then, a set of features, frequently referred to as tokens, are separated out of each source word document. The token features are given as input to Stemming. Subway tokens and casino tokens are examples of tokenization.

Stemming Stemming, also known as the reduction of inflected (or occasionally derived) words to their stem, is the act of eradicating attaches (preceeding and succeeding) from features. This approach is consumed to lessening the quantity of structures in the subspace and increase the accuracy of the categorizer by stemming the numerous types of features into a solitary feature. For instance, the stem of the words eating, eats, eaten is eat.

Term Frequency—Inverse Document Frequency (TF-IDF) The text preprocessing step takes into account every content as a feature vector, breaking down the transcript into individual terms. For TF-IDF expressions weighting, the written reports are represented as exchanges.

$$\begin{aligned} TF - IDF &= TF \times IDF \\ W_d &= f_{w,d} \times log \frac{|D|}{f_{w,d}} \end{aligned} \tag{1}$$

where $f_{w,d}$ or TF is the amount of periods 'w' seems in a transcript 'd', |D| is the dimension of the dataset, f_w, D or IDF is the amount of transcript in which 'w' performs in D in Eq. (1). A matrix representing the numerous terms and their term scores is the TF- IDF's output. The obtained term weight is given as input to stemming. Example. The word it would receive a score of 1 in the given document if it appeared 10 times and had an IDF weight of 0.1 (10*0.1 = 1). Now, the score would be 5 if the word "coffee" also occurred 10 times with an IDF weight of 0.

Pesudocode 1 Data preprocessing algorithm

```
        Function load_dataset(csv_file):
                Read CSV file into a DataFrame
                Return the DataFrame
End Function

Functuion tokenize (text):
                Split text into tokens using word_tokenize
                Filter out non-alphabetic tokens
                Convert tokens to lowercase
                Return tokens
End Function

Function stem_tokens(tokens):
                Initialize PorterStemmer
                Apply stemming to each token using the PorterStemmer
                Return stemmed tokens
End Function

Function remove_stop_words(tokens):
                Get English stop words from NLTK
                Filter out stop words from tokens
                Return filtered tokens
End Function

Function calculate_tfidf(corpus):
                Initialize TfidfVectorizer
                Transform the corpus into a TF-IDF matrix
                Get feature names from the vectorizer
                Return TF-IDF matrix and feature names
End Function

Function preprocess_sms_dataset(csv_file):
                Sms_df=load_dataset(csv_file)

                #Tokenization
                Sms_df=['tokens']= sms_df ['text'].apply(tokenize)

                #Stemming Tokens
                Sms.df ['stemmed_tokens'] = sms_df ['tokens'].apply(stem_tokens)

                #Stop words Removal
                Sms_df ['filtered_tokens'] = sms_df
['stemmed_tokens'].apply(remove_stop_words)

                #Term Frequency-Inverse Document Frequency
Tfidf_matrix, feature_names = calculate_tfidf(sms_df
['filtered_tokens'].apply(lambda x: ' '.join(x)))

#Combine TF-IDF features with the original dataset
Preprocessed_sms_df = Concatenate columns of sms_df and tfidf_matrix using
feature names
Return preprocessed_sms_df
End Function
```

As a result, this research acquired sufficient preprocessed data. Then, the preprocessed data is input to the feature removal procedure.

3.2 Feature extraction

The feature removal method creates new characteristics by linearly combining the existing content. The resultant set of characteristics will have numbers that are distinct from those of the unique features. The primary objective is for the same information to be captured with fewer features. For feature extraction, an Improved PCC-PCA is projected to overcome the computational difficulty in the existing research. This proposed method can convert low-dimensional data into high-dimensional data without information loss. Additionally, it extracts temporal data including word limit characteristics from Linguistic Inquiries, subject features, and token features. The mathematical expression (2-8) is given as follows.

The preprocessed data $X_A = \{X_{A1}, X_{A2} \ldots \ldots \ldots \ldots \ldots .X_{AN}\}$ to create the process by taking the subsequent actions: The sample number is A, and the variable number is N.

The data are in the middle, and the variance is estimated to have an average of 0 and a difference of 1.

$$\mathrm{B} = \frac{1}{a}\sum_{i=1}^{a}\phi(X_i)\phi(X_i)^L \tag{2}$$

To determine the eigenvalues and eigenvectors using the variance matrix.

$$\lambda C = BC = \frac{1}{a}\sum_{i=1}^{a}(\varnothing(\mathrm{X_i})^{\mathrm{L}}, C)\varnothing(\mathrm{X_i})\lambda C \tag{3}$$

The following formula calculates the centralized data and introduces the Gaussian kernel function.

$$K_z(x_{iz}, x_{yz}) = \exp\left(-\frac{\left|x_{iz} - x_{yz}\right|^2}{\sigma}\right) \tag{4}$$

The following formula center's the kernel matrix.

$$\overline{K_z} = K_z - 1_N K_z - K_z 1_N + 1_N K_z 1_N \tag{5}$$

In the formula, 1N is an N-dimensional square of all elements with $^1/_N$. The data with greater contribution rates are created by utilizing the following form after this research sort the eigenvalues:

$$\frac{\sum_{i=1}^{n}\lambda_i}{\sum_{1}^{n}\lambda_n} \geq 0.9 \tag{6}$$

Extract the eigenvector based on the main element. The T^2 and the SPE(squared prediction error) control limits are calculated using the following form.

$$T_a^2 = \frac{A(n^2-1)}{n(n-A)}F_{A,n-A;\alpha} \tag{7}$$

In formula,$F_{A,n-A;\alpha}$ is the critical value of the F distribution with A and n -A degrees of freedom and confidence levels of α.

$$\begin{cases} \{\delta_\alpha^2 = \theta_1\left(\frac{c_\alpha\sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0(h_0-1)}{\theta_1^2}\right)^{1/h_0} \\ \theta_i = \sum_{j=A+1}^{m}\lambda_j^i (i = 1, 2, 3) \\ h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_1^2} \end{cases} \tag{8}$$

Algorithm 2 PCC-PCA feature selection algorithm

```
Function calculate_variance(data):
    # Equation 2: Calculate the variance of the data
    variance = CalculateVariance(data)
    Return variance
End Function
Function calculate_eigen(data):
    # Equation 3: Evaluate eigenvalues and eigenvectors
    eigenvalues, eigenvectors = CalculateEigenvaluesAndVectors(data)
    Return eigenvalues, eigenvectors
End Function
Function get_gaussian_matrix(eigenvectors, eigenvalues):
    # Equation 4: Get the Gaussian Matrix
    gaussian_matrix = CalculateGaussianMatrix(eigenvectors, eigenvalues)
    Return gaussian_matrix
End Function
Function sort_eigenvalues(eigenvalues):
    # Sort eigenvalues in descending order
    sorted_eigenvalues = SortEigenvalues(eigenvalues, descending=True)
    Return sorted_eigenvalues
End Function
Function calculate_control_limits(sorted_eigenvalues):
    # Equation 7: Evaluate control limits values
    control_limits = CalculateControlLimits(sorted_eigenvalues)
    Return control_limits
End Function
Function extract_features(gaussian_matrix, control_limits):
    # Equation 8: Extract features using Gaussian Matrix and control limits
    feature_vector = ExtractFeaturesFromGaussianMatrix(gaussian_matrix,
control_limits)
    Return feature_vector
End Function
# Main function to extract relevant features
Function extract_relevant_features(preprocessed_sms_data):
    # Assume preprocessed_sms_data is a data structure containing preprocessed SMS
data
    # Calculate Variance
    variance = calculate_variance(preprocessed_sms_data)
    # If variance meets certain condition, proceed
    if variance > threshold_variance:
        # Calculate Eigenvalues and Eigenvectors
        eigenvalues, eigenvectors = calculate_eigen(preprocessed_sms_data)
        # Get Gaussian Matrix
        gaussian_matrix = get_gaussian_matrix(eigenvectors, eigenvalues)
        # Sort Eigenvalues
        sorted_eigenvalues = sort_eigenvalues(eigenvalues)
        # Calculate Control Limits
        control_limits = calculate_control_limits(sorted_eigenvalues)
        # Extract Features
        feature_vector = extract_features(gaussian_matrix, control_limits)
        Return feature_vector
    End Function
```

As a result, this research extracts the accurate features using Improved Pearson Correlation Coefficient Principal Component Analysis (PCC-PCA). Then, the extracted features are fed into the Enhanced CNN.

3.3 Enhanced Convolutional Neural Network (CNN)

For the organization procedure, Enhanced CNN is used. Thus this research uses an enhanced Convolutional Neural Network (Enhanced CNN) which is detailed in Fig. 3 to overcome the exploding gradients that have limited accuracy in the neural network. So, this research introduces Parameterized ReLU, which minimize architecture complexity, regularizing, and early stopping. Enhanced CNN layers are parameterized ReLU, max pooling, fully linked coatings, and Softmax Layer. Figure 3 depicts a convolutional neural network (CNN) in broad strokes.

Convolutional layer Numerous filters pass over the convolution operation for the incoming given information [26, 28]. The result of this tier is then calculated as the product of the filters' element-by-element multiplying and the input's receptive field. The weighted summation is one of the components in the layer below. Each convolution operation has three parameters: stride, filter size, and zero buffering. The sliding step is determined by stride, a positive integer value. The filter size for each filtering used in such a convolutional operation must be the same (receptive field). Zero-padding increases the original input matrix's rows and columns by 0 to control the size of the resulting feature space. The inclusion of the information at the edge of the input matrix is the main objective of zero padding. When there is no padding, the convolutional result is less than the input. Rectified linear unit, or ReLU, consists of the convolution operation and the activation function, which are both linear and nonlinear processes. For acquiring attributes, a particular sort of linear process known as convolution is utilised. It employs a tensor-sized assortment of variables known as the kernel, an extremely limited set of data, to the input.

Pooling layer The in-plane dimensionality of the extracted features is reduced by a pooling layer performing a characteristic downsampling process in order to provide transformation invariance to slight twists and distortions and restrict the amount of ensuing learnable parameters. Although filter size, step, as well as buffering are fixed parameters in pooling processes, alike to convolution operations, the max pooling include a trainable parameter.

Parameterized ReLU function The simplest method for parameterizing a ReLU function with a hinge-like structure is to attach a transformation function to each part of the object separately. To put it another way, let the hinge's two edges turn independently of one another. This is a p-ReLU function, as shown in Eq. (9)

$$f(a) = \begin{cases} \alpha.a \; if \; a > 0 \\ \beta.a \; if \; a \leq 0 \end{cases} \tag{9}$$

where α and β are tangible-esteemed scaling aspects of the affirmative and undesirable parts, the p-rectified linear unit (α, β) by fixing α = 1. Similarly, p-ReLU (α, 0) is alternative streamlined form of importance. Similar to the p-Sigmoid, every buried nodes i has an

effect on the function's parameters. Hence, the imitatives of p- rectified linear unit to a, α_i, and β_i for EBP are in Eq. (10–12)

$$\frac{\partial f_i(a_i)}{\partial a_i} = \begin{cases} \alpha_i \, if \, a_i > 0 \\ \beta_i \, if \, a_i \leq 0 \end{cases}, \tag{10}$$

$$\frac{\partial f_i(a_i)}{\partial \alpha_i} = \begin{cases} a_i \, if \, a_i > 0 \\ 0 \, if \, a_i \leq 0 \end{cases}, \tag{11}$$

$$\frac{\partial f_i(a_i)}{\partial \beta_i} = \begin{cases} 0 \, if \, a_i > 0 \\ a_i \, if \, a_i \leq 0 \end{cases}, \tag{12}$$

Meanwhile the principles of α_i and β_i are not controlled, the indication of a_i cannot be contingent from f_i (a_i). Thus, it is essential to retain a_i for backpropagation which outcomes in extra retention tradition. As for the p-Sigmoid case, for p- rectified linear unit U (α_i,0), a_i avoid maintaining by not informing the zero-assessmentedα_i. According to this, ramping up ReLU outcomes could result in gradient explosions, but scaling down ReLU outcomes may be able to avoid these eruptions from happening. Additionally, even if the gradient does become excessively massive for p-ReLU operations, this problem can be resolved by straightforward gradient clipping and ReLU yield price clipped techniques.

Max pooling The supreme typical pooling procedure is max pooling (MP), which outcomes the greatest benefit from every patched after separating covers from the input feature maps. In practise, MP is often taken in addition to a 2*2 filter and a step of 2. It causes the in-plane scale of map characteristics to be downsampled twice.

Fully connected layer The final convolution or pooling layer's effect characteristic maps are typically deformed or transformed into a one-dimensional (1D) sequence of numerals (or vectors) which are linked to one or more substantial layers, also known as fully connected levels, in which each input and output is connected by a learnable weight. The characteristics produced by the convolution levels and the down sampling levels are then mapped to the channel's last product, such as the chances for every category in arrangement techniques, by a selection of fully linked layers. In the end fully associated stratum, the amount of outcome nodes often corresponds to the amount of sessions. Following each completely linked layer comes a nonlinear function, like ReLU.

Softmax function The activation function in the output nodes of cnn architectures that forecast a multivariate regression chance difference is called the softmax function. In order to create a probabilistic model with K possibilities equivalent to the exponentials of the input parameters, the softmax function normalises a vector z of K actual figures. It accepts this vector as an input. Before applying softmax, certain vector elements might be negative or larger than one, which would indicate that their sum might not be 1. Even so, each element will still be in the range display style (0,1)(0,1) after applying softmax, and the elements will sum to 1, making them interpretable as chances. Additionally, a higher probability will follow from greater input components.

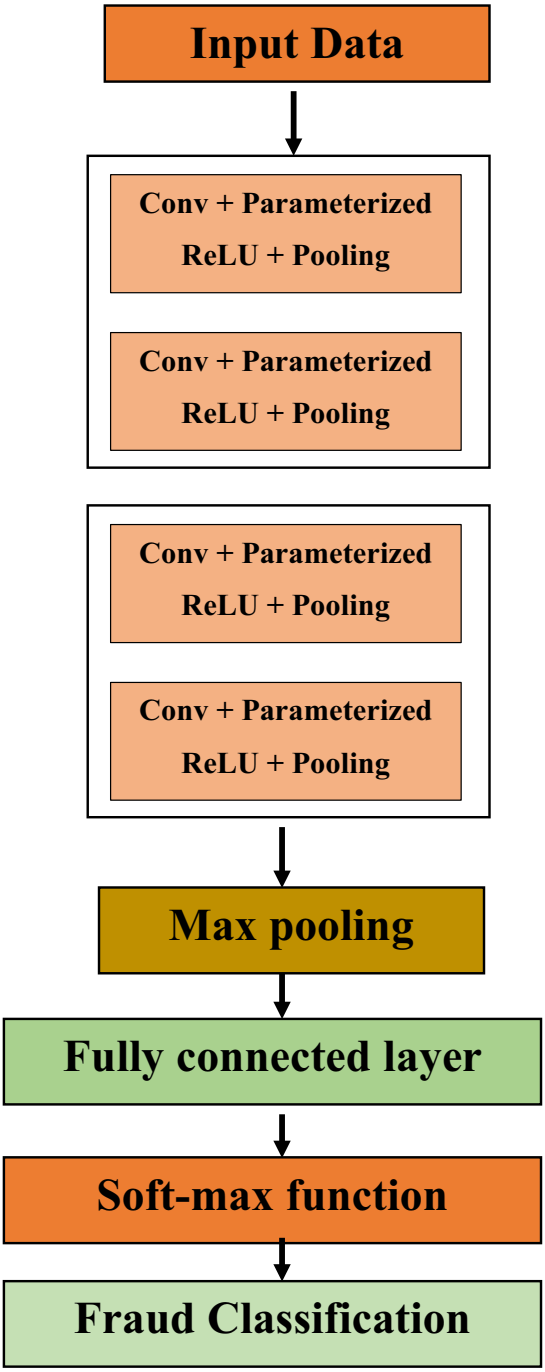

Fig. 3 Enhanced Convolutional neural network (CNN)

The softmax function $\sigma : R^K \rightarrow (0,1)^K$ is outlined when K is superior than one by the expression (13)

$$\sigma(Z)_i = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}} \text{for i} = 1, \ldots \ldots \text{.K and Z} = (Z_1 \ldots \ldots ..Z_K) \in R^K \tag{13}$$

4 Result and discussion

This chapter described the effectiveness of this offered remedy and the outcomes of its execution. Furthermore, comparison results from current studies are displayed. Here, this research utilized a tool of python, operating system is windows 7 (64-bit), intel premium processor and 8 GB RAM.

4.1 Detailed of dataset

The UCI Machine Learning Repository's SMS Spam Collection v.1 dataset was used for this study [27]. It is a group of 425 SMS spam emails that were properly culled from web pages by scanning them. 10,000 authentic texts from individuals at the same university are included in the NUS SMS Corpus. A selection of 3,375 communications, including ham Text messages, have been picked at random as shown in Fig. 4.

The Input data is split into two distinct sets during the modelling stages. They are as follows training sets and a testing set. The proportion of training to testing in the dataset is 70:30. The data for SMS Spam acquired v.1 which is created using 450 ham Text messages gathered, 1,002 ham SMS messages, and 322 spam SMS messages from Sms Messages, v.0.1 Big. As seen in Fig. 4, every row of data begins with the right class, whether ham or spam, preceded by the actual text. As a result, 4,827 occurrences of the assessed messages are classified as ham, while 747 instances are classified as spam.

4.2 Feature extraction

Length, token count, unique token count, unique token count percent, length clean, token count clean. Figure 5 Shows the feature extraction of PCC-PCA.

The assortment consists of a text archive that has the raw information on each line, followed by the appropriate category. Firstly, the message's length is determined; following that, token counts in both unique and percent are performed; then, the message is cleaned; and last, token counts are sent. For instance, spam Free Msg: Txt: Call 86,888 to receive your prize of 3 h of talk time that you may use right now, from this this research extract the features by using our proposed PCC-PCA which shown in Fig. 5.

4.3 Performance parameters

The following techniques were used to assess how well the proposed approach, Enhanced CNN, performed.

Training loss vs Validation loss How closely a deep learning algorithm fits the training data is determined by the training loss metric. The training dataset is a portion of the information that was utilised to first train the model.

To evaluate the deep neural network model's performance on the validation data, a metric known as loss of validation is used. A percentage of the information called the validation set is utilized to estimate how well the design works. The mistakes on every instance in the validation set are added to determine the validation loss, just like with the training loss.

The training and validation loss over the amount of an epoch is shown in Fig. 6. At times, the Test loss increases, whereas the training loss remains stable. From Fig. 6 the obtained training and validation loss is 0.020, 0.195 at epoch 10.

Training accuracy vs validation accuracy Figure 7 shows the accuracy of training and validation. While test accuracy refers to the training sample correctly detecting unrelated photos that were not used in training, accuracy rate refers to the utilisation of similar pictures for training and testing.

At epoch 10 the training accuracy is 99.8%, and the validation accuracy is 97.8%.

Comparision of training and validation in loss and accuracy The Fig. 8 represents the accurateness and losses. When comparing accuracy obtains higher than losses.

By using this Enhanced Convolution Neural Network, this research obtain an accuracy value is 99.8% and a loss value is 1%.

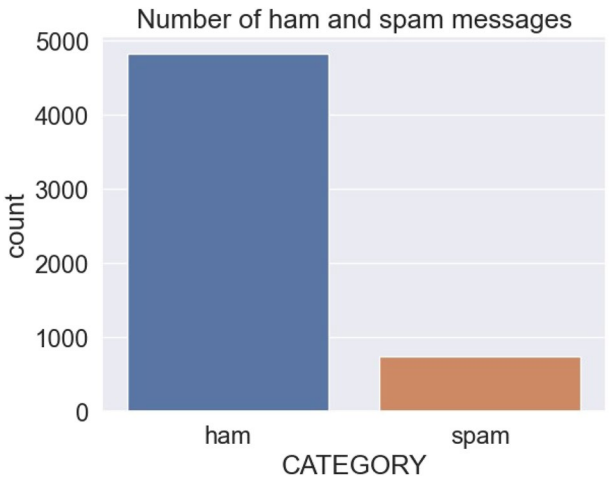

Fig. 4 Number of Spam and Ham Messages

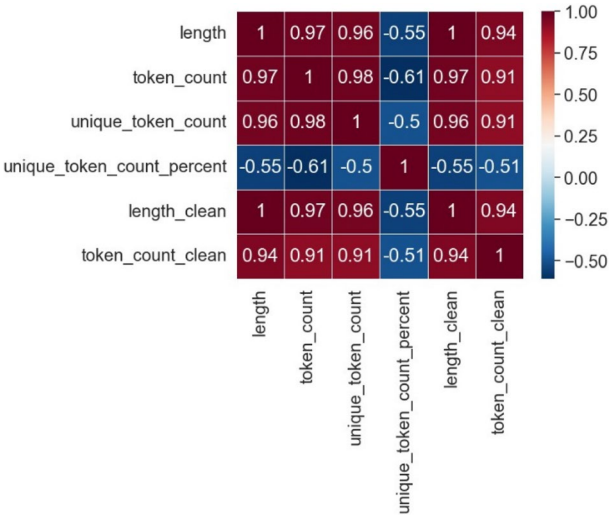

Fig. 5 Feature extraction of PCC-PCA

Confusion matrix To describe the effectiveness of a classification method, a confusion matrix is a table measurement of effectiveness The metrics used to assess selectivity, sensitization, as well as correctness are true positive and negative (TP and TN), false positive and negative (FP and FN). The number of exactly predicted scam is known as True Positive. The quantity of accurately identified non predicted scam is known as True Negatives.

In Fig. 9 the confusion matrix for the categorization of CNN is illustrated. False Positives: Instances where the model incorrectly flagged a legitimate SMS as phishing. False Negatives: Instances where the model failed to detect a phishing SMS. The diagonal values in the confusion matrix represent properly anticipated instances of a specific class. Figure 9 shows that the proposed models are highly efficient at predicting abnormalities. The predicted value is 98 and the true label is 736.

4.4 Comparison analysis

The comparison of some Proposed model metrics (Accuracy, Precision, Recall, F1score, and Kappa),R^2, MSE, RMSE, MAE. Furthermore, this newly developed method is contrasted with the existing procedure such as the SVM, NB, MNB, KNN(1), and KNN (3) are explained in the below section.

4.4.1 Comparision of performance parameters

The comparison study for Accuracy, Exactness, Retention, F1-score, and Kappa are discussed in this section. The evaluation of the proposed model metrics is illustrated in Fig. 10, which is shown below. The mathematical expression (14-18) evaluates the performance parameters of the proposed approach.

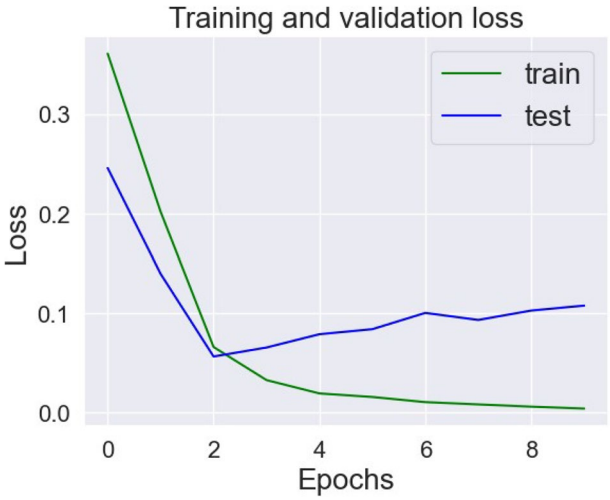

Fig. 6 Training loss and validation loss over the number of epoch

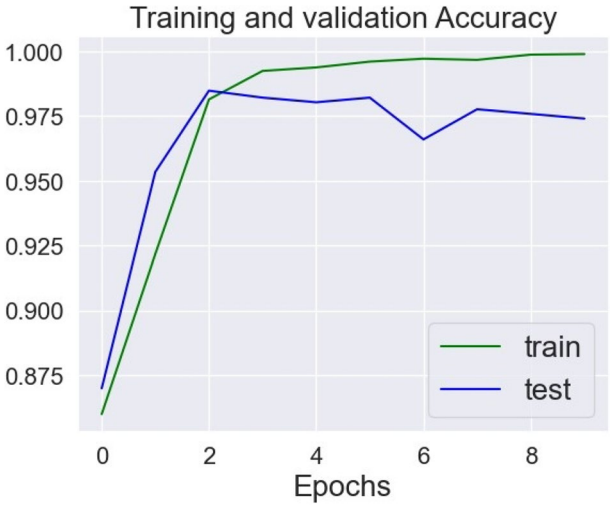

Fig. 7 Accuracy of training and testing

- **Accuracy:** The number of groupings a system accurately predicts divided by the overall amount of hypotheses utilised is the definition of accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{14}$$

A is the proportion of effectively classified texts that are categorised with precision. True Positive, which states to the ordering of texts as spam; True Negative, which denotes to the classification of texts as ham; False Positive, which discusses to the improper categorization of ham messages as spam; and False Negative, which discusses to the incorrect classification of spam messages as ham.

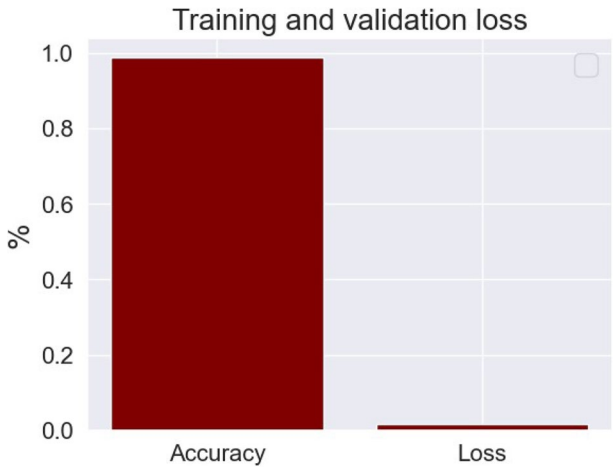

Fig. 8 Accuracy and Loss

- **Recall:** The percentage of positive observations that are correctly recognised as legitimate relative to all acceptable SMS in the information set is known as recall. It can express as

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

- **Precision:** The proportion of the set of successful samples that are accurately identified as valid to all SMS identified as valid is defined as precise. It can define

$$\text{Precision} = \frac{\text{TP}}{TP + FP} \tag{16}$$

- **F1 score:** The word F1-Score is defined as one that strikes a balance between recall and precision. It is characterised by

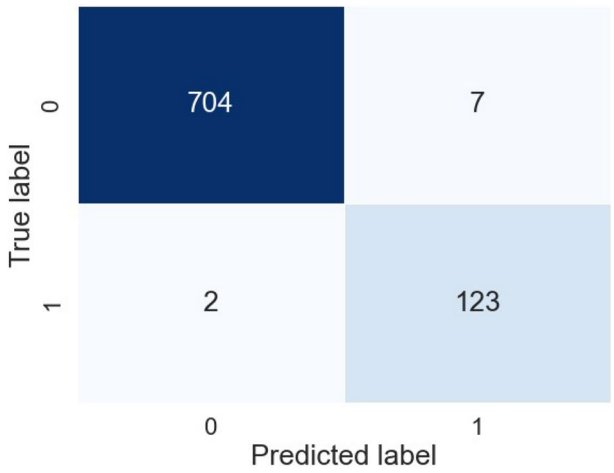

Fig. 9 The confusion matrix

$$\text{F1Score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{17}$$

- **Kappa:** The proportion of how frequently the appraisers accord to how frequently they could possibly agree is known as the kappa. It can be defined as

$$\text{Kappa} = \frac{(total\ accuracy - random\ accuracy)}{(1 - random\ accuracy)} \tag{18}$$

The value of the novel enhanced model metrics is calculated by using the above-mentioned formulas. Hence, this proposed model gives an accurateness of 99.8%, a exactness value is 98%, a Recall value is 99.75%, the F1 score value is 99.00%, and the Kappa value is 98.80% which displays the efficiency of the novel technique.

4.4.2 Error analysis

The comparative findings of R^2, MSE, RMSE, and MAE are explained in Eq. (19–22). The comparison of the enhanced CNN model metrics is displayed in Fig. 11 and is described further. The major purposes of the R^2, Mean Square Error (MSE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) metrics in regression analysis are to assess forecast failure rates and performance was evaluated.

- **MAE**, which is determined by be around the actual difference across the set of data, shows the variation among the actual and forecasted results.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \tag{19}$$

- **MSE** which is evaluated by squaring the mean variance throughout the given dataset, shows the variance among the actual and forecast values.

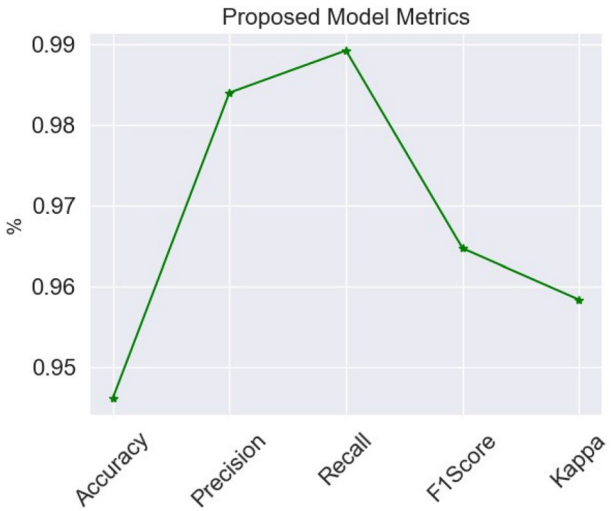

Fig. 10 Enhanced CNN metrics

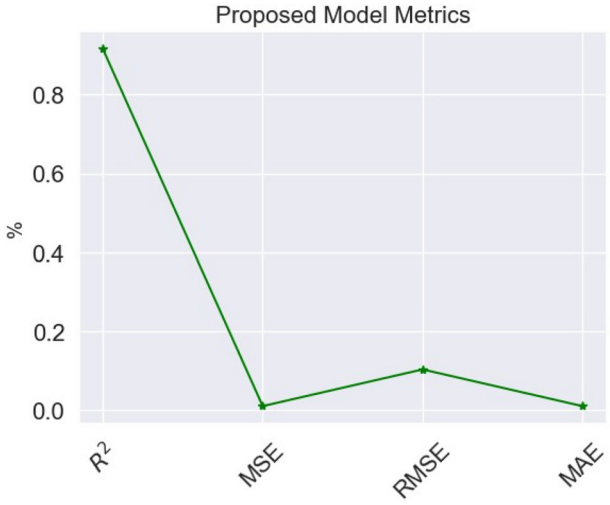

Fig. 11 Proposed model metrics

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}\right)^2 \tag{20}$$

- **RMSE** is the error rate by the root of MSE.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}\right)^2} \tag{21}$$

- **R-squared** The degree to which the factors fit perfectly in respect to the initial values is indicated by the R-squared. In proportional terms, the range 0 to 1 is converted. The value increases with higher levels of quality.

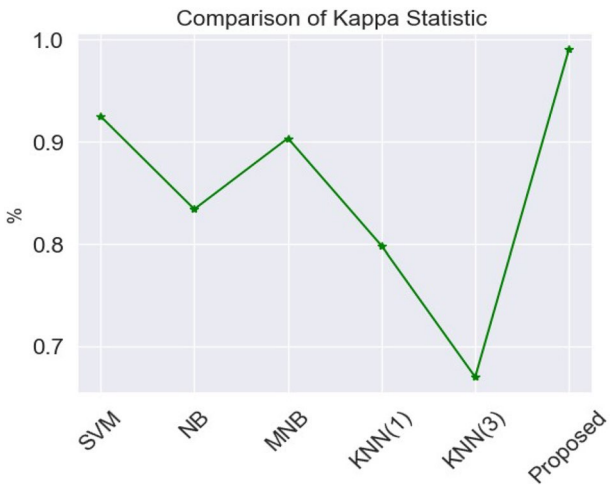

Fig. 12 Comparison of Kappa Statistics

$$R^2 = 1 - \frac{\sum \left(y_i - \hat{y}\right)^2}{\sum \left(y_i - \bar{y}\right)^2} \tag{22}$$

$\hat{y}$ = Predicted value of y, $\bar{y}$ = mean value of y.

The R^2 value is given as 99%. The MSE value is given as 1%. The Mean Absolute Error (MAE) value is given as 10%. The RMSE value is given as 1%.

4.4.3 Comparison of existing approaches

This method evaluates the proposed method's comparative results with those of this new procedure and the based approach such as the Naive Bayes (NB), Multinomial Naive Bayes (MNB), and K-Nearest Neighbor (KNN 1 and 3), and SVM. Figure 12 illustrates the overall comparison of the existing model.

The other three classifiers are connected to the proposed technique SVM, NB, MNB, and KNN with dissimilar $K = 1,3$ and 5. Figure 12 illustrates the overall comparison of the proposed technique attains higher kappa statistics by using the Deep Learning-Based SMS Fraud Resilient Model. This proposed approach compared with the baseline SVM [24], NB [24], MNB [24], K-NN (1) [24], and K-NN (3) [24] such as 92.45%, 83.37%, 90.31%, 79.79%, and 66.97%. As a result, this unique strategy outperformed current methods with a Kappa Statistics of 99.95%.

5 Conclusion

This study provides Artificial Intelligence Techniques-based Telecommunication Fraud Resilient Framework for Efficient and Accurate Detection of SMS Phishing. Pre-processing, Feature removal, and categorization are the three modules to find efficient and accurate detection of SMS phishing. The data are preprocessed and cleansed by the following methods such as tokenization, TF-IDF, and stemming. Improved Pearson Correlation Coefficient Principal Component Analysis (PCC-PCA) for feature extraction which has an advantage of low dimensional data into high dimensional data without loss in information, Moreover, this research introduces an enhanced Convolutional Neural Network (Enhanced CNN) which, overcomes the exploding gradients, this research introduces Parameterized ReLUwhich minimizes architecture complexity, regularizing, and early stopping. When compared to existing techniques, this Enhanced Convolutional Neural Network (CNN) achieves high accuracy efficiency and reduces time. The following factors were taken into account when evaluating every classifier's effectiveness: greater precision, less time taken, small loss, and the fewest FP occurrences. As a result, the proposed method is performed well and attains 99.8% percent accuracy when compared to the other technique.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Hilas CS, Mastorocostas PA (2008) An application of supervised and unsupervised learning approaches to telecommunications fraud detection. Knowl-Based Syst 21(7):721–726
2. Jain V (2017) Perspective analysis of telecommunication fraud detection using data stream analytics and neural network classification based data mining. Int J Inf Technol 9(3):303–310
3. Hila CS, Sahalos JN (2007) An application of decision trees for rule extraction towards telecommunications fraud detection. In Knowledge-Based Intelligent Information and Engineering Systems: 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 12–14, 2007. Proceedings, Part II 11 Springer Berlin Heidelberg:1112–1121. https://link.springer.com/chapter/10.1007/978-3-540-74827-4_139
4. Hilas CS, Kazarlis SA, Rekanos IT, Mastorocostas PA (2014) A genetic programming approach to telecommunications fraud detection and classification. In Proc. 2014 Int. Conf. Circuits, Syst. Signal Process. Commun. Comput pp 77–83. https://www.researchgate.net/profile/Constantinos-Hilas/publication/261191177_A_Genetic_Programming_Approach_to_Telecommunications_Fraud_Detection_and_Classification/links/0046353441f851ac2a000000/A-Genetic-Programming-Approach-to-Telecommunications-Fraud-Detection-and-Classification.pdf
5. Estévez PA, Held CM, Perez CA (2006) Subscription fraud prevention in telecommunications using fuzzy rules and neural networks. Expert Syst Appl 31(2):337–344
6. Xu W, Pang Y, Ma J, Wang SY, Hao G, Zeng S, Qian YH (2008) Fraud detection in telecommunication: a rough fuzzy set based approach. In 2008 International Conference on Machine Learning and Cybernetics IEEE 3:1249–1253. https://doi.org/10.1109/ICMLC.2008.4620596
7. Patel Y (2019) Cross channel fraud detection framework in financial services using recurrent neural networks (Doctoral dissertation, London Metropolitan University). https://repository.londonmet.ac.uk/id/eprint/6133
8. Subudhi S, Panigrahi S (2016) Use of fuzzy clustering and support vector machine for detecting fraud in mobile telecommunication networks. Int J Secure Network 11(1–2):3–11
9. Olszewski D (2012) A probabilistic approach to fraud detection in telecommunications. Knowl-Based Syst 26:246–258
10. Zhao Q, Chen K, Li T, Yang Y, Wang X (2018) Detecting telecommunication fraud by understanding the contents of a call. Cybersecurity 1:1–12
11. Ulfath RE, Alqahtani H, Hammoudeh M, Sarker IH (2021) Hybrid CNN-GRU framework with integrated pre-trained language transformer for SMS phishing detection. In The 5th International Conference on Future Networks & Distributed Systems pp 244–251. https://doi.org/10.1145/3508072.3508109
12. Mishra S, Soni D (2021) Dsmishsms-a system to detect smishing sms. Neural Comput Appl 35(7):4975–4992. https://link.springer.com/article/10.1007/s00521-021-06305-y
13. Sonowal G (2020) Detecting phishing SMS based on multiple correlation algorithms. SN Comput Sci 1(6):361
14. Noah N, Tayachew A, Ryan S, Das S (2022) Poster: PhisherCop-An Automated Tool Using ML Classifiers for Phishing Detection. In Proceedings of the 43rd IEEE Symposium on Security and Privacy (IEEE S&P 2022). https://doi.org/10.2139/ssrn.4096243
15. Sadiq A, Anwar M, Butt RA, Masud F, Shahzad MK, Naseem S, Younas M (2021) A review of phishing attacks and countermeasures for internet of things-based smart business applications in industry 4.0. Human Behavior Emerg Technol 3(5):854–864
16. Kalaharsha P, Mehtre BM (2021) Detecting Phishing Sites--An Overview. arXiv preprint arXiv:2103.12739. https://doi.org/10.48550/arXiv.2103.12739
17. Clasen M, Li F, Williams D (2021) Friend or foe: An investigation into recipient identification of SMS-based phishing. In Human Aspects of Information Security and Assurance: 15th IFIP WG 11.12 International Symposium, HAISA 2021, Virtual Event, July 7–9, 2021, Proceedings 15 Springer International Publishing:148–163. https://doi.org/10.1007/978-3-030-81111-2_13
18. Choudhary N, Jain AK (2018) Comparative analysis of mobile phishing detection and prevention approaches. In Information and Communication Technology for Intelligent Systems (ICTIS 2017) Springer International Publishing 1(2):349–356. https://doi.org/10.1007/978-3-319-63673-3_43
19. Onuodu FE, Nnaa SB (2020) An enhanced fraud detection model using neural networks for telecommunications and smart cards in nigeria. London Journal Of Research In Computer Science And Technology 20(2):27
20. Zheng YJ, Zhou XH, Sheng WG, Xue Y, Chen SY (2018) Generative adversarial network based telecom fraud detection at the receiving bank. Neural Netw 102:78–86

21. Kashir M, Bashir S (2019) Machine learning techniques for sim box fraud detection. In 2019 International Conference on Communication Technologies (ComTech) IEEE pp 4–8. https://doi.org/10.1109/COMTECH.2019.8737828
22. Boukari BE, Ravi A, Msahli M (2021) Machine learning detection for smishing frauds. In 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC) IEE pp 1–2. https://doi.org/10.1109/CCNC49032.2021.9369640
23. Hameed SM, Ali ZH (2021) SMS Spam Detection Based on Fuzzy Rules and Binary Particle Swarm Optimization. Int J Intell Eng Syst 14(2). https://doi.org/10.22266/ijies2021.0430.28
24. Wu T, Zheng KF, Wu CH, Wang XJ SMS Phishing detection using oversampling and feature optimization method
25. Sjarif NNA, Yahya Y, Chuprat S, Azmi NHFM (2020) Support vector machine algorithm for SMS spam classification in the telecommunication industry. Int J Adv Sci Eng Inf Technol 10(2):635–639
26. Kadhim AI (2018) An evaluation of preprocessing techniques for text classification. Int J Comput Sci Inf Security (IJCSIS) 16(6):22–32
27. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Networks 5(2):157–166
28. Mikolov T (2012) Statistical language models based on neural networks. Presentation at Google, Mountain View, 2nd April 80(26). https://www.fit.vutbr.cz/~imikolov/rnnlm/google.pdf
29. Srinivasarao U, Sharaff A (2023) Machine intelligence based hybrid classifier for spam detection and sentiment analysis of SMS messages. Multimed Tools Appl 1–31. https://doi.org/10.1007/s11042-023-14641-5
30. Oswald C, Simon SE, Bhattacharya A (2022) Spotspam: Intention analysis–driven sms spam detection using bert embeddings. ACM Transactions on the Web (TWEB) 16(3):1–27