IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 August 2023, accepted 22 September 2023, date of publication 26 September 2023, date of current version 5 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3319485

RESEARCH ARTICLE

Robust Deep Neural Network-Based Framework for Predicting and Classifying Capsid Protein Based on Biomedical Data

ANEES UR RAHMAN KHATTAK[1,2], AMIN ULLAH[1], AMJAD REHMAN[3], (Senior Member, IEEE), TARIQ MAHMOOD[3,4], QAMAR WAHID KHATTAK[5], SARAH ALOTAIBI[6], AND SAEED ALI OMER BAHAJ[7,8]

[1]Department of Computer Science, Faculty of Information Technology and Computer Science, University of Central Punjab, Lahore 54700, Pakistan
[2]Parsons Corporation, 87313 Riyadh, Saudi Arabia
[3]Artificial Intelligence and Data Analytics (AIDA) Laboratory, CCIS, Prince Sultan University, Riyadh 11586, Saudi Arabia
[4]Department of Information Sciences, University of Education, Vehari Campus, Vehari 61100, Pakistan
[5]Department of Physical Therapy, NCS University System, Peshawar 25000, Pakistan
[6]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 145111, Saudi Arabia
[7]MIS Department, College of Business Administration, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[8]Department of Computer Engineering, College of Engineering and Petroleum, Hadhramout University, Mukalla, Hadhramout, Yemen

Corresponding author: Saeed Ali Omer Bahaj (bahajsaeedali@gmail.com)

ABSTRACT Capsid protein is a pathogenic protein that needs to be examined because it helps in the virus's proliferation and mutation. Due to this protein, the virus can replicate and reproduce itself. The virus's outer boundary is made of capsid protein. Capsid protein analysis and prediction are essential. Several approaches, including mass spectrometry, have been developed to detect and predict Capsid protein. However, these methods are time-consuming and expensive and require highly skilled human resources. Therefore, this study proposed an efficient and robust classification approach for Capsid protein. The proposed model employs several machine learning, data science, and pattern recognition strategies to measure statistical moments based on obtained data. The experimental analysis reveals that the proposed model has achieved an overall 99% accuracy. These marks indicate that the suggested method outperformed the cutting-edge methods for classifying Capsid and non-Capsid proteins.

INDEX TERMS Capsid protein data, healthcare, bioinformatics, feature extraction, machine learning, health risks.

I. INTRODUCTION

Capsid protein is the coat or head of the virus. Its essential protein is a fragment of the compound forming the shielding shell around the nucleic acids of the virus [1]. In prokaryotic viruses, this secure shell is also denoted as the head. The Capsid surrounds the genetic substance of the virus. Capsids are divided into two types regarding their structure. Most viruses have helical or icosahedral Capsid [2].

When the virus has infected a cell, then it begins reproducing itself. Capsid subunits are produced using the protein biosynthesis of the cell. The viruses, containing those with ribonucleic acid (RNA) as a genetic material, the Capsid proteins then co-assemble with their genetic material [3]. Expect of the umbra; the genetic martial is enclosed by more than one type of coating layer protein (CP) [4]. Capsid protein protects the virus genomic martial from deprivation during reproduction in the diseased plant or other body, transforming the viral genome from one organism to another. In the earlier days, however, the pure idea depended on the virus. Capsid is involved in every phase of the viral infectivity cycle, plus carriage of the virus into the organism, the transformation of viral genetic material, reproduction of the viral genome, virus drive in the plant, beginning or conquest of host battlements, and conduction of the virus to healthy plants. Current data show that different phases of the infectivity cycle are closely linked [5], [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Gyorgy Eigner.

This study collected Capsid and non-Capsid protein datasets to evaluate the efficiency of the proposed algorithm. First, this study refines and preprocesses the data to clean and remove duplication. Afterward, features are extracted and classified using the neutral network-based algorithm to find the optimal result [7], [8]. Our research introduces a novel and comprehensive approach to feature extraction from protein sequences, designed to provide a holistic understanding of sequence characteristics. The innovation involves integrating multiple techniques, including raw moments, central moments, Hahn moments, position-specific indices, and physicochemical properties, yielding diverse features that collectively contribute to a more nuanced representation of sequences. The introduction of AAPIV (Amino Acid Position-Specific Index Vector) is particularly novel, a concept that uncovers amino acid distribution patterns and potential functional roles within sequences. This, coupled with calculating various moment-based descriptors, enables the discovery of hidden structural traits. Furthermore, we go beyond traditional sequence analysis by considering both original and reverse sequence orientations, acknowledging the context-specific information each orientation provides. This practical approach is ready for application in bioinformatics analyses and offers a tangible solution for researchers seeking a comprehensive toolkit for sequence analysis. The potential applications of our work extend to drug discovery efforts, where the diverse set of features extracted from sequences could aid in identifying potential drug targets. Moreover, we anticipate our features could offer insights into sequence evolution, contributing to understanding conservation and divergence patterns. We envision extending our methodology to other biological sequence types and incorporating machine learning techniques for predictive purposes, pushing the boundaries of feature-driven sequence analysis.

Towards this end, our proposed work's major contributions are:

1) To develop a robust neural network-based approach for classifying capsid protein.
2) To develop a novel feature extraction methodology for protein sequences, providing a holistic understanding of sequence characteristics.
3) The study integrates machine learning methodologies to improve predictive analysis and feature-driven sequence analysis by incorporating raw moments, central moments, Hahn moments, position-specific indices, and physicochemical properties.
4) The study has significantly improved the accuracy of classifying capsid and non-capsid proteins through rigorous performance evaluation and benchmarking of cutting-edge methods.

An overview of this work is presented as follows. Related work, traditional method flaws, and research motivation are reviewed in Section II. Section III describes the proposed algorithms based ANN, proposed methodology, feature extraction, and Statistical Moments Calculation. Section IV discusses the suggested approach's performance and experimental results are carried out using various performance measures. Section V explains the importance of the proposed approach's performance, Implications and Applications, limitations, and future directions. Finally, Section VI conclude the research work's findings

II. RELATED WORKS

Classification of capsid protein using artificial neural networks (ANNs) has become an increasingly popular approach in recent years [9]. Capsid proteins are the outer layer of a virus, responsible for protecting and transporting the viral genome. Their classification is essential for understanding virus evolution, pathogenesis, and the development of antiviral therapies. Previously a lot of research and practical work is done. Robert et al. [10] try to solve the problem of singular Capsid proteins, which are private and must be portrayed by solid transformative protection proposed Capsid structure-based viral classification. They searched for amino acid and nucleotide sequences similar to show the affect-ability of the methodology. They recognize an up-and-comer quality for the pandora virus Capsid protein. They show that the structure-based grouping is strongly upheld by amino corrosive and nucleotide arrangement likenesses, recommending that the similitude is because of standard drop. The correspondence between structure-based and succession-based investigations of similar proteins that appeared here allows them to be utilized in future examinations of the connection between straight grouping data and macro molecular capacity, just as between direct arrangement and protein folds.

The paper introduces an improved version of the Chimp Optimization Algorithm (ChOA) called ChOA(II) and compares it with ChOA(I). These versions use different global and local search strategies for better data clustering performance. The paper incorporates seven chaotic maps to enhance ChOA's optimization capability due to its sensitivity to chaotic values. The proposed method is evaluated using benchmark and shape datasets, compared against various optimization algorithms and hybrid approaches. The paper reviews related literature, explains data clustering concepts, presents ChOA and its stages (encircling, exploitation, utilization, exploration), and highlights the role of chaotic behavior. It also discusses the Generalized Normal Distribution Algorithm (GNDA) and Opposition-Based Learning (OBL). The study aims to improve optimization for data clustering and provides experimental results for validation [6].

Jhon et al. [11] have described the Capsid protein and its role. William et al. [12] explore the protein rings topology of the icosahedral structure of Capsid protein in bacteriophage virus. The introduced study explored the evidence that poliovirus and other picornavirus particles are modified explicitly by having myristic acid covalently bound to a Capsid protein. Chow et al. [13] describe X-ray diffraction analysis of a human immunodeficiency virus (HIV-1). Capsid (CA) protein shows that each monomer within the dimer consists of seven α -helices, five of which are arranged in a spiral coil-like structure. Kirnbauer et al. [14] describe infection by certain human papillomavirus types are regarded as the significant risk factor in the development of cervical cancer, one of the most common cancers of women worldwide. Arshan et al. [15] identified and classified the Capsid protein and provided details about several Capsid or coat protein groups. Mart et al. [16] describe and explain the origins of viral Capsid protein using cellular ancestors. The suggested approach explored the different origins of viral proteins, and Capsid is one of the Viral proteins. Shanshan et al. [17] explain the structural folds of Capsid proteins, the role and functionality of the Capsid protein, and how they are incorporated with each other.

Boroujeni et al. [18] proposed a hybrid approach using Random Forest Ranking (RFR) and Binary Dragonfly Algorithm (BDA) to identify significant genes in microarray datasets. The method removes irrelevant genes and selects optimal genes, while the BDA optimizer uses a Naïve Bayes classifier. Experimental results show the hybrid approach significantly outperforms existing metaheuristic methods in classification accuracy and optimal gene selection. In a study by Zhang et al. (2020), an ANN was used to classify capsid proteins based on their amino acid sequences. The model was trained using a large dataset of known capsid protein sequences and evaluated using cross-validation. The results showed that the ANN could accurately classify capsid proteins with a high degree of accuracy, outperforming traditional machine learning methods such as support vector machines (SVMs) [19], [20]. A study by Singh et al. utilized ANNs to classify capsid proteins into their respective structural classes. The study employed a feedforward neural network with multiple hidden layers trained on a dataset of sequence-based features [21]. The model was evaluated using 10-fold cross-validation and achieved an overall accuracy of 95.7%. The results indicated that the model was highly influential in classifying capsid proteins and outperformed other methods, such as Support Vector Machines and Decision Trees [22].

In conclusion, the literature review of recent studies on the classification of capsid proteins using ANNs shows that these methods have great potential for accurately classifying capsid proteins based on their amino acid sequences, structures, and virus families. ANNs such as MLPs, CNNs, RNNs, and deep learning methods are adequate for this task, outperforming traditional machine learning methods such as SVMs [23].

III. MATERIALS AND METHODS

A. BENCHMARK DATASET

Chou's formulation was adopted to evaluate the efficacy of the proposed algorithm [24].

To facilitate clear exposition, we adopted Chou's peptide formulation [25], a widely employed technique in computational biology. This methodology has found extensive usage in various tasks, such as the prediction of signal peptide cleavage sites [26], identification of nitrotyrosine sites [27], determination of methylation sites [28], as well as the recognition of hydroxyproline and hydroxylysine sites [29], [30]. Additionally, it has been instrumental in pinpointing lysine ubiquitination sites [31], discerning protein-protein binding sites [32], predicting phosphorylation sites [33], detecting lysine succinylation sites [34], uncovering sumoylation sites [35], and exploring diverse lysine PTM sites [36].

This study selects 2399 Capsid proteins from UniProt (UNIVERSAL PROTEIN) database [37] and obtained the family number of these Capsid proteins. We selected reviewed data of 2390 positive proteins from the UniProt database and obtained 2100 reviewed harmful proteins using filters. Afterward, the identification process determines whether a Protein is a Capsid. The selection of proteins from the UniProt dataset is based on structural information such as data source and building of a negative dataset, determined by two criteria. Negative instances are selected from protein families that differ from their own and cannot be selected from positive protein families. The duplication numbers are removed using a novel algorithm and extracting a long Capsid sequence of their family's consequent to the non-duplication numbers from UNIPROT. Finally, the categorization process splits proteins into capsid and non-capsid.

B. PROPOSED METHODOLOGY

The proposed framework for recognizing and classifying protein capsids consists of several steps: data acquisition, preprocessing, fracture selection and feature extraction, and classification. Figure 1 depicts proposed framework for recognising and classifying protein capsids.

The raw dataset impacted by the distinctive impairments and noises requires more clarity, denoising, and normalization. This study acquired a Benchmark dataset for Capsid classification from a Uniprot source in the first step [38], [39]. The proposed approach selected the positive and negative capsid proteins for model evaluation. We extracted a 1567 positive Capsid and 1587 negative instance to evaluate the proposed schemes' performance. Besides, the data is preprocessed to remove duplicate numbers through novel data preprocessing techniques [40], [41]. The proposed approach enhanced classification effectiveness and mitigated over- and under-fitting issues. As a result, positive and negative datasets were cleaned from blank spaces and special characters. This study transforms imbalanced data, drastically decreasing computation time and erroneous prediction (positive and negative) [42], [43]. Then, the features are selected and retrieved from the preprocessed data for the algorithm's robust training. We extracted 1567 instances of the positive Capsid and 1587 instances of the negative Capsid to evaluate the performance of the suggested schemes. Finally, our proposed neural network training combines all relative position-based statistical moments.

C. PREPROCESSING

The collecting and processing of data, including data discretization, interpolation, and the assimilation of unbalanced

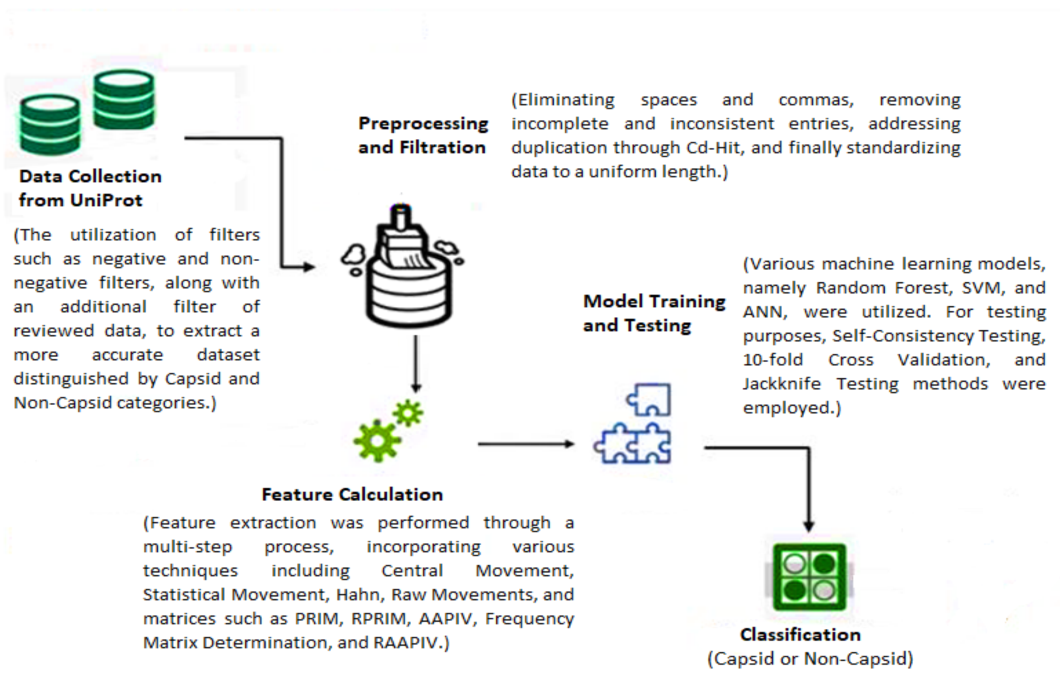

Data Collection from UniProt
(The utilization of filters such as negative and non-negative filters, along with an additional filter of reviewed data, to extract a more accurate dataset distinguished by Capsid and Non-Capsid categories.)
Preprocessing and Filtration
(Eliminating spaces and commas, removing incomplete and inconsistent entries, addressing duplication through Cd-Hit, and finally standardizing data to a uniform length.)
Model Training and Testing
(Various machine learning models, namely Random Forest, SVM, and ANN, were utilized. For testing purposes, Self-Consistency Testing, 10-fold Cross Validation, and Jackknife Testing methods were employed.)
Feature Calculation
(Feature extraction was performed through a multi-step process, incorporating various techniques including Central Movement, Statistical Movement, Hahn, Raw Movements, and matrices such as PRIM, RPRIM, AAPIV, Frequency Matrix Determination, and RAAPIV.)
Classification
(Capsid or Non-Capsid)

FIGURE 1. The figure presents a sequential workflow demonstrating the process of identifying and categorizing Capsid Proteins. This includes data collection, preprocessing, fracture calculation, model training and testing, and the final classification step.

data, is essential for effectively training the proposed models [44], [45]. In addition, the main features were identified throughout the discretization phase. The suggested technique removes noise from the database and improves the performance of classifiers [46], [47]. The redundant data is removed using a novel cluster database at the High Identity with Tolerance (CD-Hit) approach. In addition, this study used the CD-HIT-EST method to cluster similar proteins (DNAs) that meet our defined similarity threshold of 0.6 means 60%. The highly quick CD-HIT managed the huge proposed dataset. Many sequence analysis jobs may be significantly simplified with CD-HIT, which helps to comprehend the data structure and rectify biases inside datasets. Afterward, this study removes the missing data and special characters and spaces. Preprocessing drastically decreased the positive dataset from 1567 to 1207 and the negative dataset from 1587 to 793.

D. FEATURE EXTRACTION

During the preprocessing step, the CD-HIT tool [48], [49], [50] eliminates spaces, special characters, and duplicate data from the positive and negative datasets. The final preprocessed benchmark dataset consists of 1207 positive and 793 negative samples. Capsid samples using Chou's schema [24] are expressed as in Equation 1.

$$P(\xi)(\mathbb{C})=R_{-\xi}R_{-(\xi-1)}\ldots R_{-2}R_{-1}\mathbb{C}R_{+1}R_{+2}\ldots R_{+(\xi-1)}R_{+\xi} \qquad (1)$$

Here, the double line highlights the significance of amino acids.

In Equation 1, T is the subscript ξ called an integer, R_+ is the upstream filtrate from the midpoint, and R_- is the downstream filtrate. Classification of P into two classes is given by:

$$P_{\xi(T)} = \begin{Bmatrix} P_{\xi}^{+}(T) \\ P_{\xi}^{-}(T) \end{Bmatrix} \qquad (2)$$

The benchmark datasets comprise a training and testing dataset for statistical prediction. The former is for model training, and the latter is for testing. A separate benchmark dataset is not required when using jackknife for model prediction and cross-validation sub-sampling, as different combinations of dataset test outcomes are obtained. This study reduced the benchmark dataset to a statistical estimate, such as the training and testing dataset. When a prediction model is tested using K-fold cross-validation, the obtained results will be accurate due to different independent dataset combinations.

The optimal value for ξ is 20, which results in a sample consisting of $2\xi + 1 = 41$ residues, as shown in Equation 1.

$$S = C^{+} \cup C^{-} \qquad (3)$$

In Equation 3, C^+ contains 1567 positive samples, while C^- consists of 1587 negative data. The symbol $\cup$ represents the "union of sets" operation.

In biological sequence analysis, the main challenge lies in finding a suitable representation of data that retains the pattern information and characteristics necessary for target analysis. While vector formulation or detached models are often used for machine learning algorithms, they risk losing crucial sequence information. To address this, Chou's Pseudo Amino Acid Composition (PseAAC) was proposed to capture sequence patterns effectively.

In the past, Pseudo Amino Acid Composition (Pessac) was introduced for protein pattern sequences. Chou's Pseudo Amino Acid Composition has found application across various protein computation domains. Consequently, three prominent software tools were developed for public access: Pessac-General3, Pessac-Builder, and Prop. The latter two focus on different modes of calculating Chou's Pseudo amino acids, while the first one, Pessac-General, is tailored for functional domains. Moreover, Pseudo Amino Acid implemented Pseudo Couple Nucleotide Composition (PseKNC) for vector feature extraction in RNA/DNA sequence analysis, extending Chou's general PseAAC concept [24].

Thus, considering Equation 1 and the PseAAC sequence concept in sequence analysis, the samples in structural layers can be formulated as:

$$P_{\xi 7(K)} = \left[\Psi_1 \quad \Psi_2 \quad \ldots \quad \Psi_n\right]^T \tag{4}$$

In Equation 4, Ψ with subscripts $1, 2, \ldots, n$ is used for feature extraction employing relative order, and T represents the transpose operator. Equation Equation1 can be modified as follows:

$$P = R_1 R_2 R_3 R_4 R_5 \ldots R_n \tag{5}$$

In Equation 5, R_i represents one of the 20 amino acids, where i can take values from 1 to 20. To simplify, numeric codes $1, 2, 3, \ldots, 20$ represent the 20 amino acids based on their single-letter codes. The components and dimensions of Equation 4 are described using the sequence of statistical moments approach.

E. STATISTICAL MOMENTS CALCULATION

The proposed study utilizes statistical moments to gather quantitative data that illustrate various properties of the dataset. Different orders of moments offer insights into data size, distribution, and tendencies. These moments are mathematically described based on polynomials and distribution functions, providing valuable information about the dataset's characteristics.

The calculations involve various moment orders, including raw, central, and Hahn moments. Raw moments address location and scale variance, aiding in mean and probability distribution asymmetry computations. Central moments, being location-independent, are computed by subtracting the mean. Hahn moments, based on Hahn Polynomials, consider location and scale variances. All these moments offer sensitivity to the dataset's order.

The formulation of raw moments is represented as:

$$M_{ij} = \sum_{q=1}^{n} p^i q^j \beta_{pq} \tag{6}$$

Here, i and j correspond to levels of moments, and the calculations are performed for specific shapes denoted by $E77$, $E7$, $E7$, and $E7L$.

Central moments are calculated using:

$$n_{ij} = \sum_{q=1}^{n} (p - \bar{x})^i (q - \bar{y})^j \beta_{pq} \tag{7}$$

In Equation 7, the calculation of central moments is described. P is transformed into a 2-dimensional P' square matrix. For Hahn moments of 2-dimensional discrete data, the following equation is used:

$$H_{ij} = \sum_{q=1}^{N-1} \sum_{p=1}^{N-1} \beta_{ij} h_i(q - N) h_j(p, N) \tag{8}$$

Determination of Position Relative Incidence Matrix (PRIM)

The mathematical model developed for protein prediction based on the primary sequence and residue relative position requires quantifying amino acid relative positions. This leads to the creation of the Position Relative Prevalence Matrix (PRIM), a 20 × 20 matrix designed for data extraction [51]. It illustrates the occurrence of amino acid pairs relative to each other within the protein sequence:

$$S_{\text{prim}} = \begin{bmatrix} S_{1-1} & S_{1-2} & S_{i-j} & S_{1-1} \\ S_{2-1} & S_{2-2} & S_{2-j} & S_{2-20} \\ S_{i-1} & S_{i-2} & S_{i-j} & S_{2-20} \\ S_{N-1} & S_{N-2} & S_{N-j} & S_{N-20} \end{bmatrix} \tag{9}$$

The sum of position-relative incidences of the j^{th} sequence residue in Equation 9 highlights the total occurrences of the j^{th} residue relative to the i^{th} residue. The key frequency for each residue is denoted by $q9 \rightarrow F$. This yields a matrix with 400 coefficients, reducing redundancy.

Determination of Reverse Position Relative Incidence Matrix (RPRIM)

To address the challenge of identifying hidden structures within capsid primary sequences that share similar protein sequences, the Reverse Position Relative Prevalence Matrix (RPRIM) is introduced. This 20 × 20 matrix also comprises 400 coefficients [51]. Similar to the PRIM concept, RPRIM calculates the prevalence of reversed relative positions for a given residue. The Reverse PRIM (EPRIM) can be represented as follows:

$$S_{\text{prim}} = \begin{bmatrix} S_{1-1} & S_{1-2} & S_{i-j} & S_{1-1} \\ S_{2-1} & S_{2-2} & S_{2-j} & S_{2-20} \\ S_{i-1} & S_{i-2} & S_{i-j} & S_{2-20} \\ S_{N-1} & S_{N-2} & S_{N-j} & S_{N-20} \end{bmatrix} \tag{10}$$

Frequency Matrix Determination

A frequency matrix represents the presence of amino acids in the sequence arrangement. This matrix is calculated as shown in Equation 11, where the frequency of each amino acid $t_1, t_2, \ldots, t_{20}$ is recorded:

$$\mathcal{L} = (t_1, t_2, \ldots, t_{20}) \tag{11}$$

This frequency matrix captures the distribution of amino acids and their occurrences in the sequence.

Accumulative Absolute Position Incidence Vector (AAPIV) Generation

While the matrix frequency serves for mining compositional data, it does not provide information about relative positions. Hence, the Accumulative Absolute Position Incidence Vector (AAPIV) is introduced. It summarizes the sum of normalized values for each amino acid based on its position in the main sequence, as in Equation 12.

$$K = (\mu_1, \mu_2, \mu_3, \ldots, \mu_{20}) \tag{12}$$

where μ_i is computed as in Equation 13.

$$\mu_i = \sum_{k=1}^{n} pk \tag{13}$$

For deep mining and nuanced statistics related to the relative AAPIC of residues in the sequence, the Reverse Accumulative Absolute Position Incidence Vector (RAAPIV) is generated. RAAPIV is created using RPRIM and transforms AAPIV from the reverse collection. It is expressed as Equation 14.

$$A = \left[n_1, n_2, n_3, \ldots, n_{20}\right] \tag{14}$$

F. THE PROPOSED OPERATION ALGORITHM

1) RANDOM FOREST

Random forests / random decision forests are an ensemble-learning technique for regression, classification, and other errands that functions by building a crowd of the decision trees at the training time and testing the class that is the mode of mean prediction and classes of the individual trees.

The Figure 2 illustrates the architecture RF algorithm, and the details of the RF classifier are shown in Table 1.

TABLE 1. Parameters of the random forest classifier.

Parameters	Values	Parameters	Values
Estimators	50	Bootstrap	TRUE
Criterion	Gina	Obscure	TRUE
max_depth	8	Jobs	1
Verbose	0	Random state	0
Warm start	TRUE		

2) ARTIFICIAL NEURAL NETWORK

In Artificial Neural Network, neurons are connected, and the preceding Neuron output is used as the following neuron input. In the stimulation unit, all past weighted input sum is added for the aggregate of inclination value added to sum "Bias value" and conversion made as exposed. A Standard data set consists of positive and negative instances. Computation of feature vector for all data sets every feature vector consists of Hahn, Raw, and Central Moments for the illustration of the 2-dimensional arrangement of primary Capsid sequences such as PRIM and RPRIM. Position relative and extraction of composition information in a Frequency matrix. In the end, a characteristic consisting of vector 133+2r factors is formed.

FIM is created by combining the characteristic Vectors such that every row corresponds to a single sample. Estimated "output matrix" created supervised that will conform to a Positive or Negative class. Feature Input Matrix used for "Neural network" training. Feature input matrix used for neural network input. Equation of motion (EOM) is used for computing errors for learning using Backpropagation. The earlier formed benchmark dataset had both positive and negative samples. A feature vector computed from all the composed samples, all feature vector central, contained raw and Hahn moments for a 2D symbol of protein primary (sub-) structure, PRIM, and RPRIM. Likewise, apart from that, the arrangement and positional info got in the form of the Frequency Matrix (FM), AAPIV, and RAAPIV added into the feature vector. Resultantly we get a feature vector covering 133+2r elements. When the entire feature vectors are joined together so that each row agrees to a single sample, a Feature Input Matrix (FIM) is built resultantly. The parameter is tuned according to the article [52], [53] [54]. An Expected Output matrix built in a supervised manner, which followed to class (negative or positive) of the consistent element in FIM. These media (EOM and FIM) aimed to train an artificial neural network. The FIM has trimmed to the input of the neural network, whereas the EOM calculates errors for knowledge through backpropagation methodology [53], [55]. Figure 3 depicts the architecture of the artificial neural network Classifier for the proposed prediction model.

3) SUPPORT VECTOR MACHINE

SVM are supervised learning models with the associated learning algorithm that analyzes data for regression analysis and classification. SVM is normally used in classification problems SVMs based on finding a hyperactive plane that greatest divides a dataset into two classes, as shown Figure 4. The details of the RF classifier have shown in Table 1. Support vectors are the data points nearest to the hyperactive plane, the points of a data set that, if removed, would alter the position of the hyperactive dividing plane. Because of this, they have considered the critical elements of a data set.

4) GRADIENT DECENT AND ADAPTIVE LEARNING

Gradient descent was adopted for Artificial neural network training which is used for minimizing error and also for calculation change rate to straight results. such as

$$\Theta = \Theta - YV_{\Theta}F(\Theta) \tag{15}$$

Objective function F(N)parameterized by N R^d. whereas the Gradient function is given as ΔΘ F(N) where y is the learning rate of the algorithm. Algorithm functioning depends upon the learning rate. The learning rate must consist of the best possible values because it takes more time if The learning rate is low, and a high"learning rate" can be caused fluctuation of functions. On the algorithm performance The adaptive learning algorithm concedes learning rate variation. Comparison of two successive iterations errors if an error increased in one iteration. The parameters for this

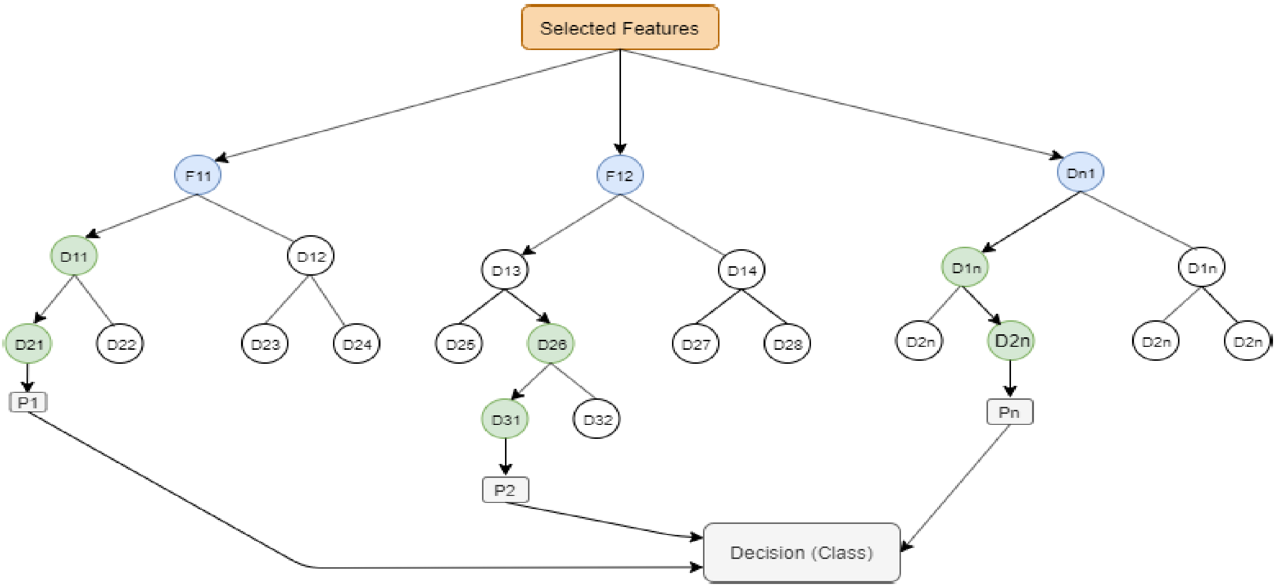


FIGURE 2. Architecture of random forest classifier for proposed prediction model.

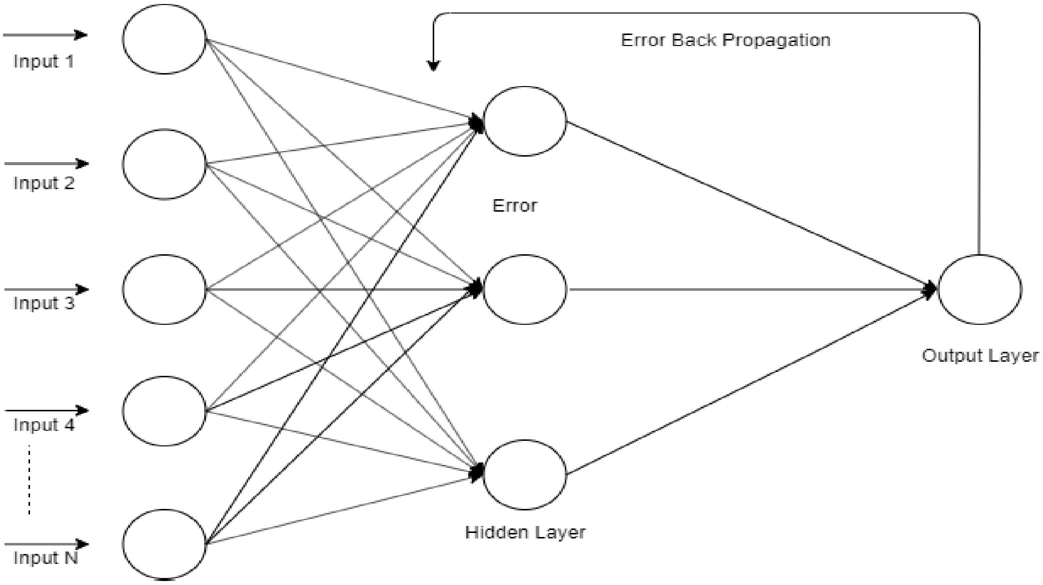


FIGURE 3. Architecture of artificial neural network classifier for the proposed prediction model.

iteration are not needed and "learning rate" will vary for minimizing function. Variation of learning rate will be on every epoch and for every successive epoch parameters are $(\Theta_0\Theta_1\Theta_2\Theta_3 \ldots..\Theta_n)$ of each epoch computed as:

$$\Theta_{m+1} = \Theta_m - Y_m VF(\Theta_M) \tag{16}$$

In Eq.18, Y_m is used as a learning rate mth epochs.

IV. RESULTS

A. ANTICIPATED ACCURACY

An essential process for budding a new prediction method is accurately calculating its accepted success rate. So, for this we have two issues. (1) Which metrics can be used for a prediction quality/? (2) Which test approach can be used for a metrics score? So, we have some metrics for the solution to the first issue (1) Sn for sensitivity, (2) Sp for specificity, (3) MCC for stability (4) Acc for accuracy. We have used Kfold cross-validation for testing.

B. METRICS FORMULATION

For a quality prediction measurement, some matrices are used, such as: (1) "ACC" used for all-over predictor accuracy measurement (2) "MCC" used for predictor stability measuring (3) Sensitivity(SN), and (4) Specificity(SP). Unluckily, there is trouble understanding old calculations, and more experiments require MCC. Here we have adopted

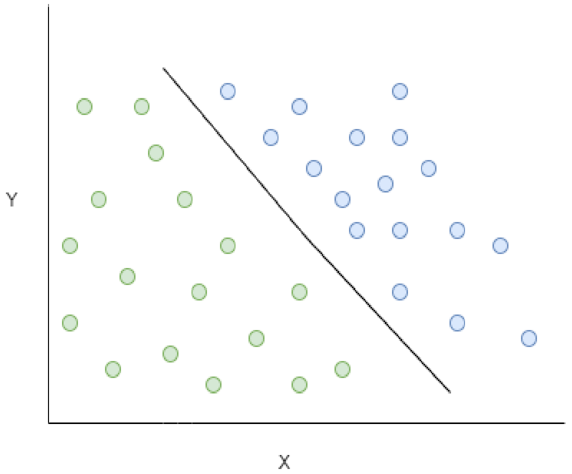


FIGURE 4. Architecture of SVM classifier for the proposed prediction model.

the techniques proposed in [56] and [57] based on the chou symbols introduced at Xu et al. and Chen et al. [25], [58].

$$S_n = 1 - \frac{N_-^+}{N^+} \quad 0 \le S_n \le 1 \tag{17}$$

$$S_p = 1 - \frac{N_-^+}{N^+} \quad 0 \le S_p \le 1 \tag{18}$$

$$Acc = 1 - \frac{(N_-^+ + N_-^+)}{N^+N^-} \quad 0 \le Acc \le 1 \tag{19}$$

$$Mcc = 1 - \frac{(\frac{N_-^+ + N_-^+}{N^+N^-})}{\sqrt{(1 + \frac{N_-^+ + N_-^+}{N^-})(1 + \frac{N_-^+ + N_-^+}{N^-})}} \quad 0 \le Mcc \le 1 \tag{20}$$

Here, N^+ = Total amount of true Capsid

N_-^+ = Count of true Capsid (mistakenly predicted to be of non-Capsid)

N^- = total number of the non-Capsid

N_+^- = Count of non-Capsid predicted mistakenly to be of Capsid.

According to Eq.11, When

$N_+^- = 0$ It shows none of the genuine Capsid that is mistakenly determined to be of non-Capsid, and the sensitivity $S_n = 1$.

When

$$N_+^- = N^+ \tag{21}$$

Shows that the genuine Capsid is mistakenly determined to be of non-Capsid, and $S{-}n = 0$. When

$$N_+^- = 0 \tag{22}$$

That's mean None of the non-Capsid is mistakenly determined as Capsid, the $S_p = 1$; Whereas $N_+^- = N^-$ meaning that all the non-Capsids are mistakenly determined to be of true Capsid, we have the Sp = 0. When $N_+^- = N_-^+ = 0$ meaning it predicated positive as positive and negative as negative, so accuracy Acc = 1 and MCC = 1; when $N_+^- = N^-$ and $N_+^- = N^-$ means total negative and positive are wrongly predicated, so accuracy Acc will be 0 and MCC will be −1. when $N_+^- = \frac{N^-}{2}$ and $N_-^+ = \frac{N^+}{2}$

Acc = 0.5 and MCC = 0 That shows it's just like a random guess. Eq.17 gives the accurate meaning of SN, SP, and accuracy, which is easier to understand, especially the meaning of MCC. The equations set defined in Eq.19 is suitable for identifying predictor quality. Method performance was calculated by calculating the prediction's sensitivity, specificity, accuracy, and MCC. The formulae for calculating these parameters [52], [59] are as follows:

where

$$S_n = TP/TP + FP \tag{23}$$

Sp uses for specificity calculated by use of this formula:

$$S_p = TN/TN + FN \tag{24}$$

Accuracy calculated by use of this formula:

$$Acc = TP + TN/TP + FP + TN + FN \tag{25}$$

Mcc is used for stability,

$$Mcc = \frac{(TP * TN)(FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FN)(TN + FP)}} \tag{26}$$

C. TEST METHOD

The test method is used for the score of four matrices. Three methods can be used for model testing by use of statistical analysis, which is the following (1) Independent data set test (2) K-fold cross-validation test (3) Self-consistency test (4) Jackknife Testing.

D. RANDOM FOREST CLASSIFIER

Random forests or random decision forests are an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual tree.

1) SELF-CONSISTENCY TESTING

Self-consistency testing method used for training and testing similar datasets. This method is commonly used for data types where the true positive value is already known. This testing result is shown in Table 2. This shows the complete PseAAC performance and shows PseACC is highly proficient, less time-consuming, and needs less manpower for implementation The results for self-consistency by random forest are shown in Table 2 while ROC has shown in Figure 5a, whose accuracy is 98.25%. Table 2 self-consistency testing via random forest classifier.

2) CROSS VALIDATION

In cross-validation, the dataset is divided into separate k-folds in which k is kept constant. For each partition, testing

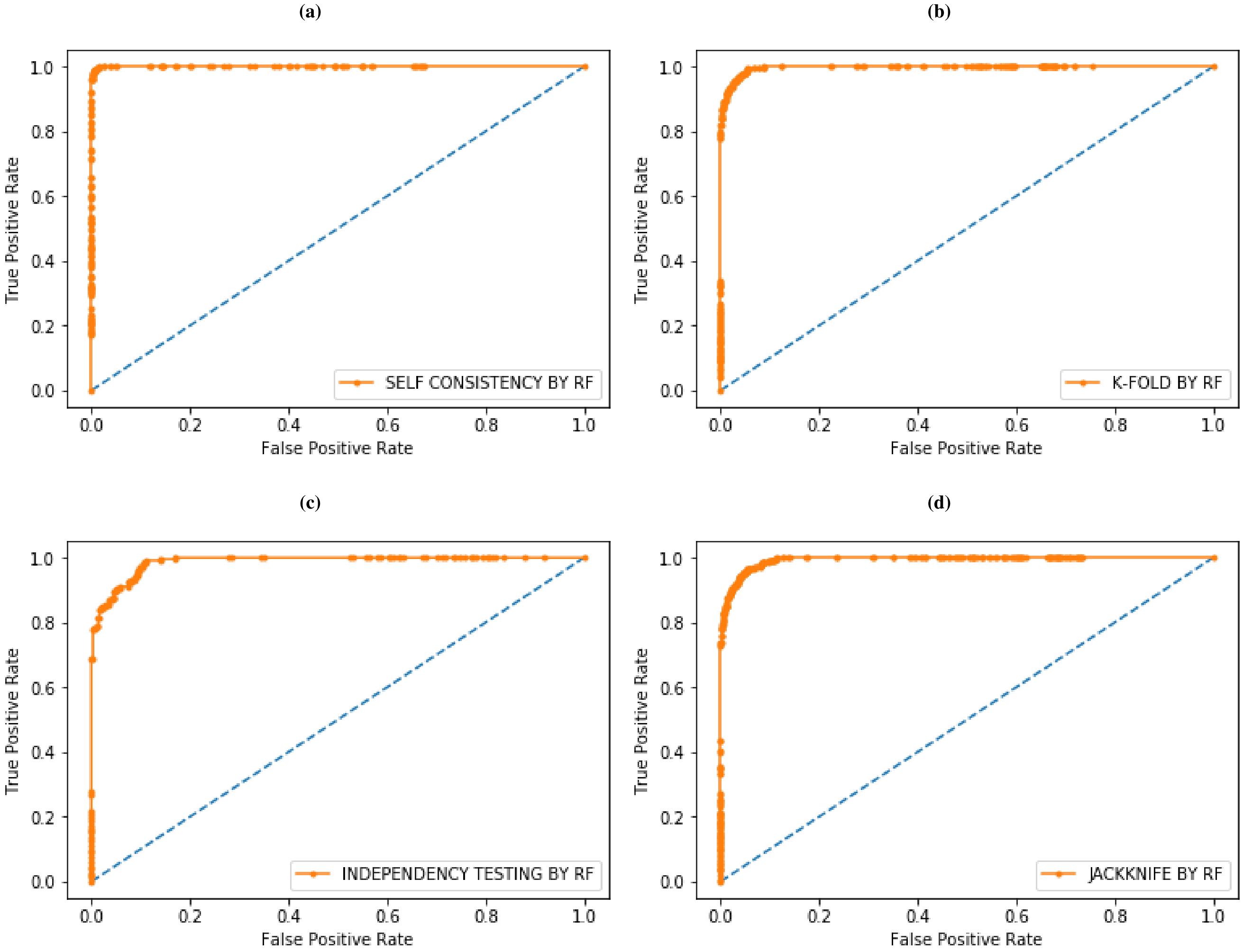


FIGURE 5. (a) ROC curve for self-consistency via random forest classifier b). ROC curve for 10-fold CV via random forest classifier, (c) ROC curve for independent results via random forest classifier, (d) ROC curve for jackknife results via random forest classifier.

TABLE 2. Self-consistency via random forest classifier.

TN	FP	FN	TP	Accuracy	Precision	Recall	F1 Score
775	18	3	1204	98.25%	0.9854	0.9975	0.9914

was performed K-times after dataset training, and accuracy was computed for each iteration. All accuracies average are described as the result of cross-validation. For positive and negative datasets, the same methodology was applied. Randomly data selection was performed to form a subset for K = 10. The results for cross-validation by random forest are shown in Table 3 while ROC is shown in Figure 5b, which accuracy is 95.80%.

3) INDEPENDENT TESTING

Independent dataset testing is done by performing a 70-30 split on the original dataset. RF classifier was trained through the 70% dataset and was tested using the remaining 30% dataset. The results for independent testing by random forest are shown in Table 4 while ROC has shown in Figure 5c, which accuracy is 93.00%.

4) JACKNIFE TESTING

In jackknife testing, each time, the model was trained on N-1, where N represents the total number of benchmark dataset instances, and testing is done by one benchmark dataset instance. Each time data is selected randomly for training and testing, and the training and testing of the model were done according to the dataset. The results for jackknife testing by random forest are shown in Table 5 while ROC is shown in Figure 5d, which has an accuracy of 94.24%.

E. ARTIFICIAL NEURAL NETWORK

The artificial neural network is a structure of linked neurons in which the last neuron's output is the next neuron's input.

1) SELF-CONSISTENCY

Self-consistency testing method used for training and testing similar datasets. This method is commonly used for data types

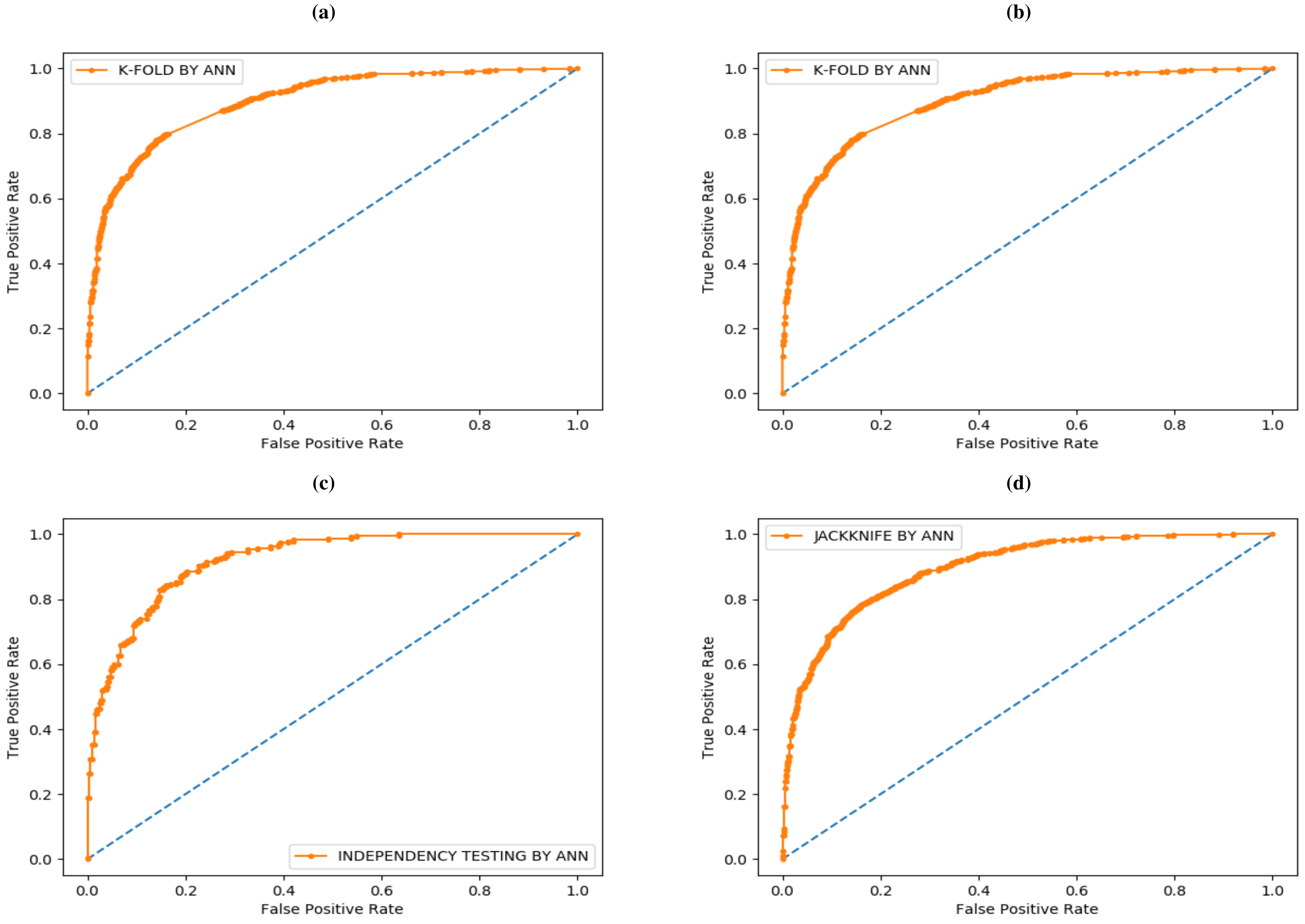


FIGURE 6. (a) ROC curve for self-consistency via ANN b). ROC curve for 10-fold CV via ANN, (c) ROC curve for independent results via ANN, (d) ROC curve for jackknife results via ANN.

TABLE 3. Cross validation result via random forest classifier.

K-Fold	TN	FP	FN	TP	Accuracy	Precision	Recall	F1-Score
1Fold	72	08	0	121	96.02	0.9375	1.0000	0.9673
2Fold	70	10	1	120	94.53	0.9231	0.9917	0.9563
3Fold	73	07	2	119	95.52	0.9447	0.9834	0.9637
4Fold	75	04	3	118	96.5	0.9677	0.9750	0.9714
5Fold	73	06	4	117	95.0	0.9516	0.9661	0.9588
6Fold	72	07	1	120	96.0	0.9459	0.9917	0.9682
7Fold	74	05	1	120	97.0	0.9608	0.9917	0.9760
8Fold	76	03	2	118	97.49	0.9756	0.9834	0.9795
9Fold	69	10	2	118	93.97	0.9216	0.9834	0.9516
10Fold	73	06	2	118	95.98	0.9516	0.9834	0.9672
	Final 10CV Score				95.80	0.9497	0.9833	0.9662

where the true positive value is already known. This testing result is shown in Table 6. This shows the complete PseAAC performance and that PseACC is highly proficient, less time-consuming, and requires less manpower for implementation. Results for self-consistency by Artificial Neural Network are shown in table 6 while ROC is shown in Figure 6a which accuracy is 82.25%.

2) CROSS VALIDATION

In cross-validation, the dataset is divided into separate k-folds in which k is kept constant. For each partition, testing was performed K-times after dataset training, and for each iteration, accuracy was computed. All accuracies average are described as a result of cross-validation. For positive and negative datasets, the same methodology was applied.

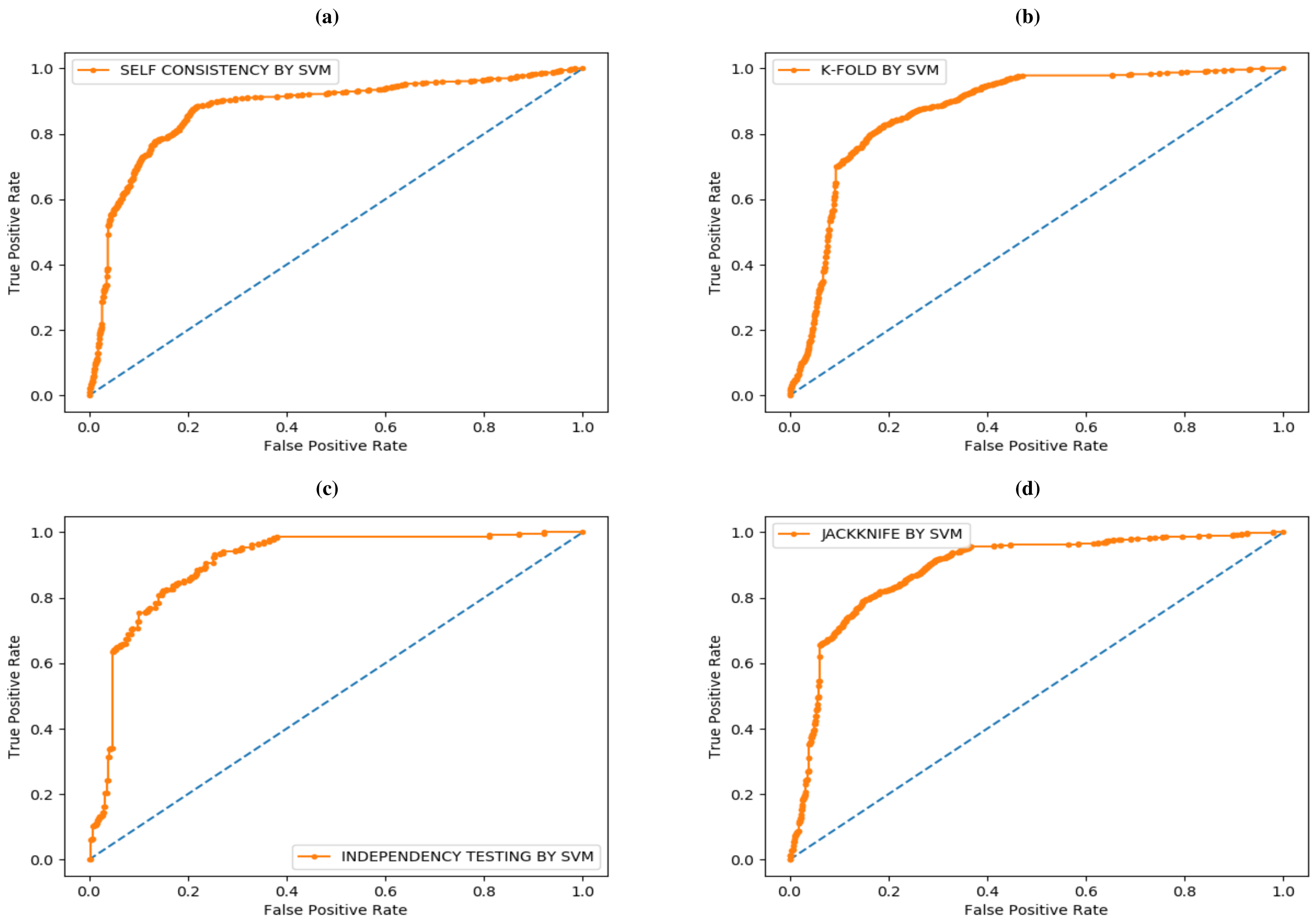


FIGURE 7. (a) ROC curve for self-consistency via SVM b). ROC curve for 10-fold CV via SVM, (c) ROC curve for independent results via SVM, (d) ROC curve for jackknife results via SVM.

TABLE 4. Independent testing via random forest classifier.

Training Dataset		Testing Dataset	
TN	518	TN	210
FP	40	FP	25
FN	16	FN	17
TP	826	TP	348
Accuracy	96.00	Accuracy	93.00
Precision	0.9538	Precision	0.9326
Recall	0.9812	Recall	0.9533
F1 Score	0.9673	F1 Score	0.9429

TABLE 5. Jackknife testing via random forest classifier.

TN	FP	FN	TP	Accuracy	Precision	Recall	F1 Score
715	78	37	1170	94.24	0.9372	0.9691	0.9529

Randomly, data selection was performed to form a subset for K = 10. Results for cross-validation by Artificial Neural Network are shown in Table 6b, while ROC is shown in Figure 6c, with an accuracy is 82.20%.

TABLE 6. Self-consistency via artificial neural network.

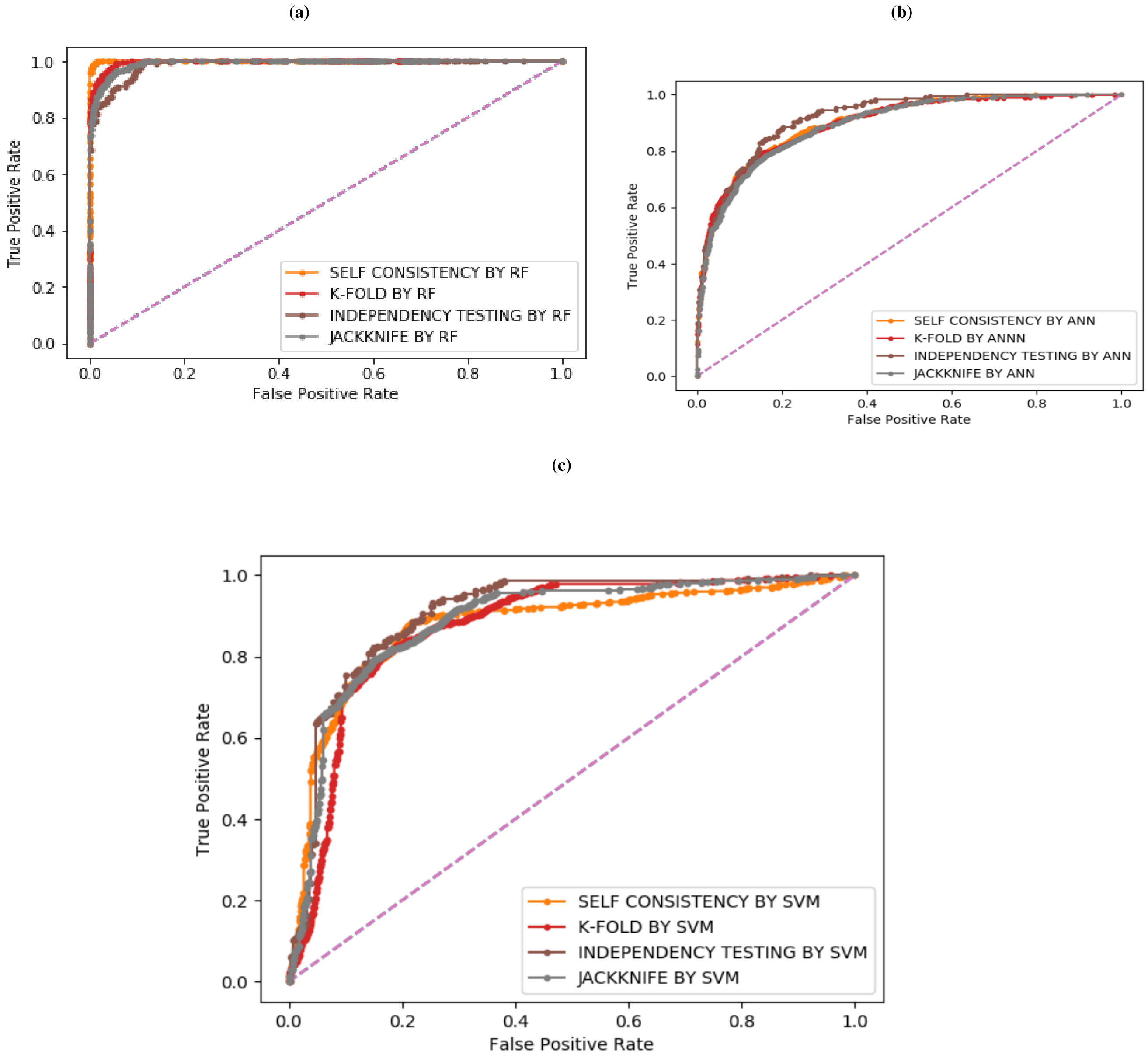
TN	FP	FN	TP	Accuracy	Precision	Recall	F1 Score
595	198	157	1050	0.8225	0.8415	0.8696	0.8553

3) INDEPENDENT TESTING

Independent dataset testing is done by performing a 70-30 split on the original dataset. RF classifier was trained through 70% dataset and tested using the remaining 30% dataset. Results for independent testing by Artificial Neural Network are shown in Table 8, while ROC is shown in Figure 6d, which accuracy is 82.83%.

4) JACKKNIFE TESTING

In jackknife testing, each time, the model was trained on N-1, where N represents the total number of benchmark dataset instances, and testing was done by one benchmark dataset instance. Data is selected randomly for training each time, and the model is tested according to the dataset. Results for jackknife testing by Artificial Neural Network are shown in Table 9, while ROC is shown in Figure 6d, which accuracy is 81.67%.


(a)

(b)

(c)

FIGURE 8. (a) ROC curve for random forest classifier b). ROC curve for SVM, (c) ROC curve for ANN.

F. SUPPORT VECTOR MACHINE

Support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.

1) SELF-CONSISTENCY

Self-consistency testing method used for training and testing similar datasets. This method is commonly used for data types where the true positive value is already known. This testing result is shown in Table 10. This shows the complete PseAAC performance and that PseACC is highly proficient, has less time consumption, and needs less manpower for implementation. Results for self-consistency by Support Vector Machine are shown in Table 10, while ROC is shown in Figure 7a, with an accuracy is 82.45%.

2) CROSS VALIDATION

In cross-validation, the dataset was divided into separate k-folds in which k was kept constant. For each partition, testing was performed K-times after dataset training, and for each iteration, accuracy was computed. All accuracies average are described as the result of cross-validation. For positive and negative datasets, the same methodology was applied. Randomly data selection was performed to form a subset for K = 10. Results for cross-validation by Support Vector Machine are shown in Table 11, while ROC is shown in Figure 7b, with accuracy is 80.75%.

Independent Testing: Independent dataset testing is done by performing a 70-30 split on the original dataset. RF classifier was trained through the 70% dataset and was tested using the remaining 30% dataset. Results for independent testing by

TABLE 7. Cross validation result artificial neural network.

K-Fold	TN	FP	FN	TP	Accuracy	Precision	Recall	F1-Score
1Fold	56	24	13	108	81.59	0.8182	0.8927	0.8544
2Fold	56	24	20	101	78.11	0.8089	0.8342	0.8214
3Fold	63	17	14	107	84.58	0.8634	0.8842	0.8737
4Fold	66	13	20	101	83.5	0.8851	0.8342	0.8589
5Fold	58	21	17	104	81.0	0.8322	0.8590	0.8454
6Fold	54	25	10	111	82.5	0.8164	0.9170	0.8645
7Fold	60	19	15	106	83.0	0.8485	0.8768	0.8624
8Fold	62	17	15	105	83.92	0.8602	0.8759	0.8680
9Fold	51	28	10	110	80.90	0.7971	0.9160	0.8521
10Fold	58	21	13	107	82.90	0.8350	0.8917	0.8624
Final 10CV Score					82.20	0.8394	0.8745	0.8567

TABLE 8. Independent testing via artificial neural network.

Training Dataset		Testing Dataset	
TN	410	TN	183
FP	148	FP	52
FN	100	FN	51
TP	742	TP	314
Accuracy	82.29	Accuracy	82.83
Precision	0.8333	Precision	0.8575
Recall	0.8810	Recall	0.8602
F1 Score	0.8565	F1 Score	0.8588

TABLE 9. Jackknife testing via multi-layer perceptrone.

TN	FP	FN	TP	Accuracy	Precision	Recall	F1 Score
587	206	157	1050	81.67	0.8357	0.8696	0.8523

TABLE 10. Self-consistency testing via support vector machine.

TN	FP	FN	TP	Accuracy	Precision	Recall	F1 Score
583	210	141	1066	0.8245	0.8357	0.8829	0.8586

Support Vector Machine are shown in Table 12, while ROC is shown in Figure 7c, which accuracy is 81.83

3) JACKNIFE TESTING

In jackknife testing, each time, the model was trained on N-1, where N represents the total number of benchmark dataset instances, and testing was done by one benchmark dataset instance. Data is selected randomly each time, and the model's training and testing are done according to the dataset. Results for independent testing by Support Vector Machine are shown in Table 13, while ROC is shown in Figure 7d, which accuracy is 81.99%.

G. COMPARISON

These validations show that the proposed predictor is an accurate and efficient way of identifying Capsid proteins based on their protein sequences. An overview of all results is shown in Figures 8c, 8a and 8b.

V. DISCUSSION

The present study focused on developing a prediction model for identifying Capsid proteins based on their protein sequences. The proposed methodology involved the utilization of several machine learning classifiers, including Random Forest, Artificial Neural Network (ANN), and Support Vector Machine (SVM), to achieve accurate predictions. The results obtained through various testing methods were assessed regarding sensitivity, specificity, accuracy, and Matthew's Correlation Coefficient (MCC), providing a comprehensive evaluation of the proposed model's performance. The accuracy of the prediction model was systematically evaluated using multiple validation techniques, including self-consistency, cross-validation, and jackknife testing. These techniques provided insights into the model's ability to generalize and perform well on unseen data. The achieved accuracy rates of 94.76%, 95.44%, and 97.38% through self-consistency, cross-validation, and jackknife testing underscore the proposed approach's robustness and effectiveness. These results imply that the model's predictions are consistent and reliable across diverse datasets. Comparative analysis of the three classifiers employed revealed exciting patterns in their performance. The Random Forest classifier demonstrated high accuracy, with an average of 95.80% accuracy through cross-validation, while the SVM and ANN classifiers achieved 80.75% and 82.20% average accuracy, respectively. Although all classifiers displayed promising performance, the Random Forest classifier consistently outperformed the others across various testing methodologies. This finding aligns with previous studies highlighting the effectiveness of ensemble methods like Random Forest in classification tasks. The successful development of an accurate Capsid protein prediction model has implications for various areas within bioinformatics and biomedicine. Accurate identification of Capsid proteins holds significant value in understanding human diseases and their underlying molecular mechanisms. With this prediction model, researchers can expedite the identification

TABLE 11. Cross validation via support vector machine.

K-Fold	TN	FP	FN	TP	Accuracy	Precision	Recall	F1-Score
1Fold	68	12	31	90	78.61	0.8824	0.7434	0.8067
2Fold	62	18	28	93	77.11	0.8371	0.7681	0.8018
3Fold	70	10	26	95	82.09	0.9048	0.7853	0.8404
4Fold	70	9	27	94	82.00	0.9122	0.7765	0.8393
5Fold	69	10	34	87	78.00	0.8969	0.7180	0.7972
6Fold	61	18	19	102	81.50	0.8505	0.8439	0.8472
7Fold	67	12	22	99	83.00	0.8919	0.8189	0.8544
8Fold	69	10	24	96	82.91	0.9057	0.8000	0.8491
9Fold	67	12	23	97	82.41	0.8891	0.8085	0.8467
10Fold	63	16	24	96	79.90	0.8571	0.8000	0.8276
Final 10CV Score					80.75	0.8802	0.7829	0.8285

TABLE 12. Independent testing via support vector machine.

Training Dataset		Testing Dataset	
TN	458	TN	202
FP	100	FP	33
FP	100	FP	33
FN	139	FN	76
TP	703	TP	289
Accuracy	82.93	Accuracy	81.83
Precision	87.5	Precision	86.79
Recall	83.4	Recall	82.22
F1 score	85.4	F1 score	84.50

TABLE 13. Jackknife testing via support vector machine.

TN	FP	FN	TP	Accuracy	Precision	Recall	F1 Score
639	154	207	1000	81.99	86.62	82.87	84.71

process, facilitating more focused investigations into disease-related processes. Furthermore, the proposed methodology can be a foundation for enhancing existing computational tools for studying protein sequences. Its accuracy and versatility make it a potential candidate for integration into bioinformatics pipelines, enabling researchers to efficiently analyze protein sequences and derive meaningful insights. Additionally, the model's robust performance across multiple validation techniques suggests its potential for generalization to other protein prediction tasks. Despite the promising results obtained, this study is not without limitations. The prediction model's performance heavily depends on the quality and diversity of the training dataset. It is crucial to continuously update and expand the dataset for new protein sequences and variations. The model's generalizability to different species and protein families should also be further investigated. Refining the model's architecture and incorporating additional features in future research could further enhance its accuracy. Exploring hybrid models that combine the strengths of multiple classifiers may lead to even more robust predictions. Moreover, investigating interpretability techniques for these machine learning models could provide insights into the biological features driving the predictions. Consequently, this study presents a novel prediction model for identifying Capsid proteins using machine learning classifiers. The achieved accuracy rates through comprehensive validation methods underscore the model's effectiveness and potential applications in bioinformatics. The comparative analysis of classifiers highlights the strengths of the Random Forest classifier while also shedding light on areas for future research. The developed model holds promise in understanding human diseases and improving the efficiency of protein sequence analysis in various biological contexts.

VI. CONCLUSION

In the realm of studying human diseases, the identification of Capsids plays a pivotal role. Our objective is to enhance the accuracy of Capsid prediction. To evaluate the accuracy of the proposed model, we have employed a comprehensive validation approach, including Jackknife cross-validation. The results of our study demonstrate PROMISING accuracy achieved through Jackknife testing, cross-validation, and self-consistency, yielding percentages of 94.76%, 95.44%, and 97.38%, respectively.

In essence, our proposed model can potentially refine outcomes in the context of uninterrupted protein sequences. Given the rapid proliferation of protein residue arrangements, there exists a pressing need for robust bioinformatics methodologies. Interdisciplinary techniques that focus on Capsid identification are pivotal in advancing our understanding of human diseases. This accuracy enhancement in prediction, achieved systematically, underscores the significance of our approach.

Our study employs an Artificial Neural Network for Capsid prediction, employing a series of steps encompassing self-consistency, cross-validation, and statistical moments. By evaluating Capsid performance, we have demonstrated the efficacy of this methodology. The results unequivocally establish that our approach contributes to improved computational outcomes for protein sequences of Capsid Proteins.

A. OUTLOOK

The focus of this study lies in the classification of Capsid and Non-Capsid proteins. However, the implications of our findings extend beyond this scope and hold promise for broader applications. Our proposed model can serve as a versatile tool that transcends its current purpose, offering opportunities for protein analysis and classification diversification.

CONFLICT OF INTEREST

The authors declare that they have no Conflict of interest.

ACKNOWLEDGMENT

The authors would like to thank Prince Sultan University for their support. This research is supported by the Artificial Intelligence & Data Analytics Lab (AIDA) CCIS Prince Sultan University, Riyadh, Saudi Arabia. The authors are thankful for the support.

REFERENCES

[1] M. Tu, F. Liu, S. Chen, M. Wang, and A. Cheng, "Role of capsid proteins in parvoviruses infection," *Virology J.*, vol. 12, no. 1, pp. 1–8, Dec. 2015.

[2] J. Rana, J. L. S. Campos, G. Leccese, M. Francolini, M. Bestagno, M. Poggianella, and O. R. Burrone, "Role of capsid anchor in the morphogenesis of Zika virus," *J. Virology*, vol. 92, no. 22, Nov. 2018, Art. no. e01174.

[3] F. An, B. T. Sayed, R. M. R. Parra, M. H. Hamad, R. Sivaraman, Z. Zanjani Foumani, A. A. Rushchitc, E. El-Maghawry, R. M. Alzhrani, S. Alshehri, and K. M. AboRas, "Machine learning model for prediction of drug solubility in supercritical solvent: Modeling and experimental validation," *J. Mol. Liquids*, vol. 363, Oct. 2022, Art. no. 119901.

[4] I. Zurnic Bönisch, L. Dirix, V. Lemmens, D. Borrenberghs, F. De Wit, F. Vernaillen, S. Rocha, F. Christ, J. Hendrix, J. Hofkens, and Z. Debyser, "Capsid-labelled HIV to investigate the role of capsid during nuclear import and integration," *J. Virology*, vol. 94, no. 7, Mar. 2020, Art. no. e01024.

[5] C. Ko, R. Bester, X. Zhou, Z. Xu, C. Blossey, J. Sacherl, F. W. R. Vondran, L. Gao, and U. Protzer, "A new role for capsid assembly modulators to target mature hepatitis B virus capsids and prevent virus infection," *Antimicrobial Agents Chemotherapy*, vol. 64, no. 1, pp. 1110–1128, Dec. 2019.

[6] S. Pedram Haeri Boroujeni and E. Pashaei, "A hybrid chimp optimization algorithm and generalized normal distribution algorithm with opposition-based learning strategy for solving data clustering problems," 2023, *arXiv:2302.08623*.

[7] Z. Ahmad, J. Li, and T. Mahmood, "Adaptive hyperparameter fine-tuning for boosting the robustness and quality of the particle swarm optimization algorithm for non-linear RBF neural network modelling and its applications," *Mathematics*, vol. 11, no. 1, p. 242, Jan. 2023.

[8] M. Marani, M. Soltani, M. Bahadori, M. Soleimani, and A. Moshayedi, "The role of biometric in banking: A review," *EAI Endorsed Trans. AI Robot.*, vol. 2, pp. 1–15, Aug. 2023.

[9] S. Iqbal, A. N. Qureshi, A. Ullah, J. Li, and T. Mahmood, "Improving the robustness and quality of biomedical CNN models through adaptive hyperparameter tuning," *Appl. Sci.*, vol. 12, no. 22, p. 11870, Nov. 2022.

[10] R. M. Sinclair, J. J. Ravantti, and D. H. Bamford, "Nucleic and amino acid sequences support structure-based viral classification," *J. Virology*, vol. 91, no. 8, Apr. 2017, Art. no. e02275.

[11] J. F. Bol, "Role of capsid proteins," in *Plant Virology Protocols*. Cham, Switzerland: Springer, 2008, pp. 21–31.

[12] W. R. Wikoff, L. Liljas, R. L. Duda, H. Tsuruta, R. W. Hendrix, and J. E. Johnson, "Topologically linked protein rings in the bacteriophage HK97 capsid," *Science*, vol. 289, no. 5487, pp. 2129–2133, Sep. 2000.

[13] M. Chow, J. F. E. Newman, D. Filman, J. M. Hogle, D. J. Rowlands, and F. Brown, "Myristylation of picornavirus capsid protein VP4 and its structural significance," *Nature*, vol. 327, no. 6122, pp. 482–486, Jun. 1987.

[14] R. Kirnbauer, F. Booy, N. Cheng, D. R. Lowy, and J. T. Schiller, "Papillomavirus L1 major capsid protein self-assembles into virus-like particles that are highly immunogenic," *Proc. Nat. Acad. Sci. USA*, vol. 89, no. 24, pp. 12180–12184, Dec. 1992.

[15] A. Nasir and G. Caetano-Anollés, "Identification of capsid/coat related protein folds and their utility for virus classification," *Frontiers Microbiology*, vol. 8, p. 380, Mar. 2017.

[16] M. Krupovic and E. V. Koonin, "Multiple origins of viral capsid proteins from cellular ancestors," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 12, pp. E2401–E2410, Mar. 2017.

[17] S. Cheng and C. L. Brooks, "Viral capsid proteins are segregated in structural fold space," *PLoS Comput. Biol.*, vol. 9, no. 2, Feb. 2013, Art. no. e1002905.

[18] S. P. H. Boroujeni and E. Pashaei, "A novel hybrid gene selection based on random forest approach and binary dragonfly algorithm," in *Proc. 18th Int. Conf. Electr. Eng., Comput. Sci. Autom. Control (CCE)*, Nov. 2021, pp. 1–8.

[19] S. N. Amrun, J. J. L. Tan, N. Y. Rickett, J. A. Cox, B. Lee, M. J. Griffiths, T. Solomon, D. Perera, M. H. Ooi, J. A. Hiscox, and L. F. P. Ng, "TREM-1 activation is a potential key regulator in driving severe pathogenesis of enterovirus A71 infection," *Sci. Rep.*, vol. 10, no. 1, p. 3810, Mar. 2020.

[20] T. Saba, A. Rehman, and S. Roy, *Prognostic Models in Healthcare: AI and Statistical Approaches*. Cham, Switzerland: Springer, 2022.

[21] S. Ali, J. Li, Y. Pei, R. Khurram, K. U. Rehman, and T. Mahmood, "A comprehensive survey on brain tumor diagnosis using deep learning and emerging hybrid techniques with multi-modal MR image," *Arch. Comput. Methods Eng.*, vol. 29, no. 7, pp. 4871–4896, Nov. 2022.

[22] V. Rajput, T. Minkina, B. Ahmed, S. Sushkova, R. Singh, M. Soldatov, B. Laratte, A. Fedorenko, S. Mandzhieva, E. Blicharska, J. Musarrat, Q. Saquib, J. Flieger, and A. Gorovtsov, "Interaction of copper-based nanoparticles to soil, terrestrial, and aquatic systems: Critical review of the state of the science and future perspectives," in *Reviews of Environmental Contamination and Toxicology*, vol. 252. 2020, pp. 51–96.

[23] M. Yaqub, F. Jinchao, K. Arshid, S. Ahmed, W. Zhang, M. Z. Nawaz, and T. Mahmood, "Deep learning-based image reconstruction for different medical imaging modalities," *Comput. Math. Methods Med.*, vol. 2022, Jun. 2022, Art. no. 8750648.

[24] H. Mohabatkar, "Prediction of cyclin proteins using Chou's pseudo amino acid composition," *Protein Peptide Lett.*, vol. 17, no. 10, pp. 1207–1214, Oct. 2010.

[25] K.-C. Chou, "Using subsite coupling to predict signal peptides," *Protein Eng., Des. Selection*, vol. 14, no. 2, pp. 75–79, Feb. 2001.

[26] H.-B. Shen and K.-C. Chou, "Signal-3L: A 3-layer approach for predicting signal peptides," *Biochem. Biophys. Res. Commun.*, vol. 363, no. 2, pp. 297–303, Nov. 2007.

[27] Y. Xu, X. Wen, L.-S. Wen, L.-Y. Wu, N.-Y. Deng, and K.-C. Chou, "iNitro-Tyr: Prediction of nitrotyrosine sites in proteins with general pseudo amino acid composition," *PLoS One*, vol. 9, no. 8, Aug. 2014, Art. no. e105018.

[28] W.-R. Qiu, X. Xiao, W.-Z. Lin, and K.-C. Chou, "iMethyl-PseAAC: Identification of protein methylation sites via a pseudo amino acid composition approach," *BioMed Res. Int.*, vol. 2014, May 2014, Art. no. 947416.

[29] Y. Xu, X. Wen, X.-J. Shao, N.-Y. Deng, and K.-C. Chou, "IHyd-PseAAC: Predicting hydroxyproline and hydroxylysine in proteins by incorporating dipeptide position-specific propensity into pseudo amino acid composition," *Int. J. Mol. Sci.*, vol. 15, no. 5, pp. 7594–7610, May 2014.

[30] W.-R. Qiu, B.-Q. Sun, X. Xiao, Z.-C. Xu, and K.-C. Chou, "iHyd-PseCp: Identify hydroxyproline and hydroxylysine in proteins by incorporating sequence-coupled effects into general PseAAC," *Oncotarget*, vol. 7, no. 28, pp. 44310–44321, Jul. 2016.

[31] W.-R. Qiu, X. Xiao, W.-Z. Lin, and K.-C. Chou, "iUbiq-Lys: Prediction of lysine ubiquitination sites in proteins by extracting sequence evolution information via a gray system model," *J. Biomolecular Struct. Dyn.*, vol. 33, no. 8, pp. 1731–1742, Aug. 2015.

[32] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "Identification of protein–protein binding sites by incorporating the physicochemical properties and stationary wavelet transforms into pseudo amino acid composition," *J. Biomolecular Struct. Dyn.*, vol. 34, no. 9, pp. 1946–1961, Sep. 2016.

[33] W.-R. Qiu, X. Xiao, Z.-C. Xu, and K.-C. Chou, "iPhos-PseEn: Identifying phosphorylation sites in proteins by fusing different pseudo components into an ensemble classifier," *Oncotarget*, vol. 7, no. 32, pp. 51270–51283, Aug. 2016.

[34] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "pSuc-Lys: Predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach," *J. Theor. Biol.*, vol. 394, pp. 223–230, Apr. 2016.

[35] J. Jia, L. Zhang, Z. Liu, X. Xiao, and K.-C. Chou, "pSumo-CD: Predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general PseAAC," *Bioinformatics*, vol. 32, no. 20, pp. 3133–3141, Oct. 2016.

[36] W.-R. Qiu, B.-Q. Sun, X. Xiao, Z.-C. Xu, and K.-C. Chou, "iPTM-mLys: Identifying multiple lysine PTM sites and their different types," *Bioinformatics*, vol. 32, no. 20, pp. 3116–3123, Oct. 2016.

[37] *UniProt*. Accessed: Jan. 2023. [Online]. Available: https://www.uniprot.org

[38] J. Amin, M. Sharif, M. Yasmin, T. Saba, and M. Raza, "Use of machine intelligence to conduct analysis of human brain data for detection of abnormalities in its cognitive functions," *Multimedia Tools Appl.*, vol. 79, nos. 15–16, pp. 10955–10973, Apr. 2020.

[39] T. Mahmood, J. Li, Y. Pei, F. Akhtar, M. U. Rehman, and S. H. Wasti, "Breast lesions classifications of mammographic images using a deep convolutional neural network-based approach," *PLoS One*, vol. 17, no. 1, Jan. 2022, Art. no. e0263126.

[40] K. U. Rehman, J. Li, Y. Pei, A. Yasin, S. Ali, and T. Mahmood, "Computer vision-based microcalcification detection in digital mammograms using fully connected depthwise separable convolutional neural network," *Sensors*, vol. 21, no. 14, p. 4854, Jul. 2021.

[41] T. Saba, M. A. Khan, A. Rehman, and S. L. Marie-Sainte, "Region extraction and classification of skin cancer: A heterogeneous framework of deep CNN features fusion and reduction," *J. Med. Syst.*, vol. 43, no. 9, p. 289, Sep. 2019.

[42] T. Mahmood, J. Li, Y. Pei, F. Akhtar, A. Imran, and K. U. Rehman, "A brief survey on breast cancer diagnostic with deep learning schemes using multi-image modalities," *IEEE Access*, vol. 8, pp. 165779–165809, 2020.

[43] A. Norouzi, M. S. M. Rahim, A. Altameem, T. Saba, A. E. Rad, A. Rehman, and M. Uddin, "Medical image segmentation methods, algorithms, and applications," *IETE Tech. Rev.*, vol. 31, no. 3, pp. 199–213, May 2014.

[44] M. Soleimani, Z. Forouzanfar, M. Soltani, and M. J. Harandi, "Imbalanced multiclass medical data classification based on learning automata and neural network," *EAI Endorsed Trans. AI Robot.*, vol. 2, pp. 1–11, Jul. 2023.

[45] M. Soleimani and A. S. Mirshahzadeh, "Multi-class classification of imbalanced intelligent data using deep neural network," *EAI Endorsed Trans. AI Robot.*, vol. 2, pp. 1–10, Jul. 2023.

[46] A. Jabbar, S. Naseem, T. Mahmood, T. Saba, F. S. Alamri, and A. Rehman, "Brain tumor detection and multi-grade segmentation through hybrid caps-VGGNet model," *IEEE Access*, vol. 11, pp. 72518–72536, 2023.

[47] S. Naseem, T. Mahmood, T. Saba, F. S. Alamri, S. A. Bahaj, H. Ateeq, and U. Farooq, "DeepFert: An intelligent fertility rate prediction approach for men based on deep learning neural networks," *IEEE Access*, vol. 11, pp. 75006–75022, 2023.

[48] W. Li and A. Godzik, "Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, Jul. 2006.

[49] S. Iqbal, A. N. Qureshi, J. Li, I. A. Choudhry, and T. Mahmood, "Dynamic learning for imbalanced data in learning chest X-ray and CT images," *Heliyon*, vol. 9, no. 6, Jun. 2023, Art. no. e16807.

[50] A. Rehman and T. Saba, "Features extraction for soccer video semantic analysis: Current achievements and remaining issues," *Artif. Intell. Rev.*, vol. 41, no. 3, pp. 451–461, Mar. 2014.

[51] W. Hussain, Y. D. Khan, N. Rasool, S. A. Khan, and K.-C. Chou, "SPrenylC-PseAAC: A sequence-based model developed via Chou's 5-steps rule and general PseAAC for identifying S-prenylation sites in proteins," *J. Theor. Biol.*, vol. 468, pp. 1–11, May 2019.

[52] A. Ullah, S. M. Anwar, M. Bilal, and R. M. Mehmood, "Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation," *Remote Sens.*, vol. 12, no. 10, p. 1685, May 2020.

[53] A. Ullah, S. U. Rehman, S. Tu, R. M. Mehmood, Fawad, and M. Ehatisham-ul-haq, "A hybrid deep CNN model for abnormal arrhythmia detection based on cardiac ECG signal," *Sensors*, vol. 21, no. 3, p. 951, Feb. 2021.

[54] A. Ullah and S. Anwar, "One dimensional convolution neural network model for ECG arrhythmia classification," *Tech. J.*, vol. 25, no. 2, pp. 85–94, 2020.

[55] Y. D. Khan, N. Rasool, W. Hussain, S. A. Khan, and K.-C. Chou, "iPhosY-PseAAC: Identify phosphotyrosine sites by incorporating sequence statistical moments into PseAAC," *Mol. Biol. Rep.*, vol. 45, no. 6, pp. 2501–2509, Dec. 2018.

[56] Y. Xu, J. Ding, L.-Y. Wu, and K.-C. Chou, "iSNO-PseAAC: Predict cysteine S-Nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition," *PLoS One*, vol. 8, no. 2, Feb. 2013, Art. no. e55844.

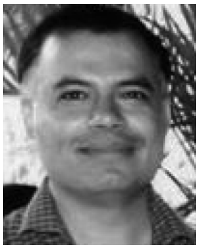
[57] W. Chen, P.-M. Feng, H. Lin, and K.-C. Chou, "iRSpot-PseDNC: Identify recombination spots with pseudo dinucleotide composition," *Nucleic Acids Res.*, vol. 41, no. 6, p. e68, Apr. 2013.

[58] K.-C. Chou, "Prediction of signal peptides using scaled window," *Peptides*, vol. 22, no. 12, pp. 1973–1979, Dec. 2001.

[59] Y. Fang, Y. Guo, Y. Feng, and M. Li, "Predicting DNA-binding proteins: Approached from Chou's pseudo amino acid composition and other specific sequence features," *Amino Acids*, vol. 34, no. 1, pp. 103–109, Jan. 2008.

ANEES UR RAHMAN KHATTAK received the B.S. degree from BUITEMS Quetta, Pakistan, and the M.S. degree from the University of Management Technology Lahore, Pakistan. He is currently working as a Senior Data Analyst at Parsons Corporation. He has more than nine years of international experience as a Senior Data Analyst, a Senior Software Engineer, a machine learning expert, and a Chief Technical Officer. He has held these positions in Pakistan's emerging IT industry organizations. He brings with him an excellent record in higher education, industrial collaboration, and administration. He is also an Official Mentor and a Panel Member of Judges for the Foundation Council of the National Incubation Center (NIC), Quetta. He has provided trainings on entrepreneurship and leadership, web development, digital marketing, and machine learning. As an Industry Expert, he was invited as a Guest Speaker by NIC, Quetta, for different sessions on entrepreneurship and leadership, digital marketing, and role of IT in business. He is an IEEE and ACM BUITEMS Chapter member who was also invited for a session on how to survive in the software industry. He has delivered multiple lectures on implementation of software engineering principles on AI algorithms and development of software for AI algorithms (PIEAS, Islamabad). As a Judge, he has attended the Foundation Council Events (2018, 2019, 2020, and 2021) NIC, Quetta and Young Leadership and Entrepreneurship Summit (YLES) Competition and the Ideation Academy Competition, in 2019, 2020, and 2021, LUMS, Lahore. His research interests include software engineering, machine learning, artificial intelligence, AI planning, bioinformatics, and image processing.

AMIN ULLAH received the B.Sc. degree in computer system engineering and the M.S. degree in electrical engineering from the University of Engineering and Technology Peshawar, in 2011 and 2014, respectively, and the Ph.D. degree from the University of Engineering and Technology Taxila, in 2020. He has been teaching for more than ten years at various renowned universities of Pakistan, before joining UCP, in September 2022. He got the Higher Education Commission (HEC), Pakistan Scholarship, in 2019, and have been a Ph.D. Research Scholar with the Computer Vision Laboratory (CRCV Laboratory), Centre for Research, University of Central Florida, Orlando, FL, USA, until 2020, under the supervision of Dr. Ulas Bagci. He is regularly publishing his research in various journals/conferences of high repute. His research contributions also include peer-reviewed conference and journal publications as an independent researcher and the coauthor. His main research interests include artificial intelligence, machine learning, deep learning, computer networks, wireless networks, and computer vision.



AMJAD REHMAN (Senior Member, IEEE) received the Ph.D. and Postdoctoral degrees (Hons.) from the Faculty of Computing, Universiti Teknologi Malaysia, with a specialization in forensic documents analysis and security, in 2010 and 2011, respectively. He is currently a Senior Researcher with the Artificial Intelligence and Data Analytics Laboratory, College of Computer and Information Sciences (CCIS), Prince Sultan University, Riyadh, Saudi Arabia. He is the author of more than 200 ISI journal articles and conferences. He is also a PI in several funded projects and also completed projects funded from MOHE Malaysia and Saudi Arabia. His research interests include data mining, health informatics, and pattern recognition. He received the Rector Award for the 2010 Best Student from Universiti Teknologi Malaysia.

TARIQ MAHMOOD received the master's degree in computer science from the University of Lahore, Pakistan, and the Ph.D. degree in software engineering from the Beijing University of Technology, China. He is currently an Assistant Professor/the HOD of the Faculty of Information Sciences, University of Education, Vehari Campus, Vehari, Pakistan. He is also a renowned expert in image processing, healthcare informatics and social media analysis, ad-hoc networks, and WSN. He is contributed with more than 35 research articles in well-reputed international journals and conferences. He is an Editorial Member and a Reviewer of various journals, including *PLOS One*, *Journal of Supercomputer*, *Journal of Digital Imaging*, *International Journal of Sensors*, and *Wireless Communications and Control*. His research interests include image processing, social media analysis, medical image diagnosis, machine learning, and data mining. He aims to contribute to interdisciplinary research of computer science and human-related disciplines.

QAMAR WAHID KHATTAK received the D.P.T. degree from the Institute of Physical Medicine and Rehabilitation (IPM&R), Khyber Medical University (KMU) in 2019. She is currently pursuing the M.S. degree in musculoskeletal physiotherapy from KMU. She has more than three years of experience in both clinical and academic settings. She worked as a Physiotherapist at Northwest General Hospital Peshawar for one year and then as a Lecturer at the Hafeez Medical Institute Peshawar and NCS University System Peshawar for two years.

SARAH ALOTAIBI received the B.Sc. and M.Sc. degrees in computer science from King Saud University, Saudi Arabia, and the Ph.D. degree in computer vision from the University of York, U.K. She is currently an Assistant Professor with the Department of Computer Science Department, King Saud University based in Riyadh, Saudi Arabia. Her research interests focus on computer vision and machine learning, more specifically: statistical modeling, appearance modeling, face modeling, reflectance analysis, inverse rendering using optimization schemes, and deep learning.

SAEED ALI OMER BAHAJ received the Ph.D. degree from Pune University, India, in 2006. He is currently an Associate Professor with the Department of Management Information Systems, College of Business Administration, Al-Kharj. He is also an Associate Professor with the Computer Engineering Department, Hadramout University, Yemen, and the MIS Department COBA, Prince Sattam Bin Abdulaziz University. His main research interests include artificial intelligence, information management, forecasting, information engineering, big data, and information security.