Expert Systems With Applications 218 (2023) 119641

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Viral genome prediction from raw human DNA sequence samples by combining natural language processing and machine learning techniques

Mohammad H. Alshayeji [a,*], Silpa ChandraBhasi Sindhu [b], Sa'ed Abed [a]

[a] *Computer Engineering Department, College of Engineering and Petroleum, Kuwait University, P.O. Box 5969, Kuwait City 13060, Safat, Kuwait*
[b] *Different Media, P.O. Box 14390, Faiha, Kuwait*

ARTICLE INFO

Keywords:
Metagenome
Machine learning
Human DNA
NLP
K-mer counting
Bag of words

ABSTRACT

Infection with a virus can lead to a range of illnesses in humans, including cancer. When viruses infect a host, they may disrupt normal host function and cause deadly diseases. Understanding complicated viral illnesses requires novel viral genome prediction. Since many of the sequences in assembled contigs from human samples are not identical to known genomes, many assembled contigs are labeled "unknown" by conventional alignments. In this study, sequences from 19 metagenomic investigations were used to create the model proposed here, and these sequences were examined and classified using BLAST. We implemented k-mer counting and the bag-of-words technique using CountVectorizer. As far as we are aware, this work represents the first framework that combines natural language processing (NLP) along with traditional ML classification approaches on raw metagenomic contigs to automatically identify viruses in a variety of human biospecimens. The suggested models are general rather than specialized to a particular viral family. Since the proposed methodology is precise and simple, we may incorporate it into computer-aided diagnosis (CAD) systems to make day-to-day hospital activities easier. In the last stage, binary classification of deoxyribonucleic acid (DNA) with normal and viral genomes was performed using traditional ML classifiers. Using the KNN classifier, the suggested model achieved 98.6% classification accuracy along with 98.5% precision, 98.6% recall, 0.984 F1 score, 0.896 Matthews correlation coefficient, 0.895 kappa, 0.97 classification success index and detection rate of 98.6% for the prediction of viral genomes in DNA. Compared to previously developed ML techniques, the model achieved a significantly greater performance for viral genome prediction.

1. Introduction

Viruses multiply and infect cells in the human body, altering metabolism in the process. Viruses can cause common infections, such as colds, warts, and flu, as well as more serious illnesses, including COVID-19 and smallpox. All viruses detected on or in the human body constitute the human virome. Human samples contain a wide variety of viruses, and the composition of these viruses has been shown to change in ill people (Liang & Bushman, 2021). Even if many viruses are discovered on a regular basis, undiscovered viruses may still exist. The hazards posed by pathogens (viruses, bacteria) are now considerably more unpredictable and challenging to control because pathogens can spread much more freely and quickly than in the past. Hence, dependable viral detection technologies are essential. One of the critical tasks of bioinformatics is detecting undiscovered viruses using experimental metagenomics records. Without any prior knowledge, biospecimens are used to obtain DNA sequences using next-generation sequencing (NGS) techniques (Meiring et al., 2012). Metagenomics represents the study of genetic material collected from a heterogeneous population of species. Genetic information can be extracted from cell, tissue, DNA, blood, RNA, protein or urine samples from plants, animals, or people (Pinu et al., 2019).

Genome sequencing is the process of analyzing the DNA extracted from a sample. Presently, NCBI BLAST is used to detect possible viral genomes in human biospecimens. It uses alignment-based classification, in which sequences are aligned to publicly available genetic material that is known, and the degree of similarity between them is calculated. Metagenomic samples may include genetic material of a large number of extremely divergent viruses that lack known homologs. As a result, BLAST classifies many sequences generated by NGS technology as "unknown" (BLAST: Basic Local Alignment Search Tool., n.d.).

The current situation is that a wide variety of viruses, particularly

[*] Corresponding author.
E-mail addresses: m.alshayeji@ku.edu.kw (M.H. Alshayeji), silpa@differentmedia-kw.com (S.C. Sindhu), s.abed@ku.edu.kw (S. Abed).

https://doi.org/10.1016/j.eswa.2023.119641
Received 23 November 2022; Received in revised form 27 January 2023; Accepted 28 January 2023
Available online 1 February 2023

those that are pathogenic toward humans, are emerging daily worldwide. Recognizing and taking steps to control pathogenic viruses will take time. The process of identifying viral structures, studying these structures in detail, documenting the findings, adding the information to reference databases, etc., is time consuming. Additionally, analyzing human DNA based on a viral database may not be useful for identifying viral genomes that may have been identified recently. Hence, the development of an ML model for predicting the presence of any viral genome in human DNA is essential for early action.

In this research, we automated the process of viral genome prediction in human DNA sequences by combining both natural language processing (NLP) and machine learning (ML) methods. Input biological sequences were divided into words having fixed lengths by k-mer counting. Then, the bag-of-words technique was applied to convert the text into the ML classifier input format. To implement this, CountVectorizer was used for text-to-numeric conversion. Classifiers such as extreme gradient boosting (XGBoost), K-nearest neighbors (KNN), support vector machine (SVM), etc., were employed and fine-tuned using grid search (GS) and tenfold cross-validation. The proposed workflow outline is illustrated in Fig. 1.

The following list summarizes this paper's main contributions:

1. In this work, we used raw metagenomic contigs from diverse human samples to construct a system that is able to automatically identify the possible viral sequences present across the genome.
2. To the best of our knowledge, this is the first work that combines NLP and traditional ML classification algorithms to develop a model for viral sequence detection in raw human metagenomic contigs. The exploratory strategy we preferred in this work tries to combine NLP and ML classification techniques for text analysis and effective model training, respectively.
3. The suggested model is generalized and not specific to a particular viral family. Hence, the model will also be applicable for the prediction of recently found viruses whose sequences may not be available in the reference databases.
4. Thus far, the proposed methodology contributes a precise, simple, fast and fully automatic framework for viral genome prediction.
5. By incorporating this model into a computer-aided diagnosis (CAD) system, it can be used in day-to-day hospital applications involving viral genome prediction from DNA, and thereby early diagnosis and treatment could be achieved by reducing healthcare professionals' workload.

The paper is structured as follows: Section 2 summarizes related research, while the third section details the database, techniques used and so on. Section 4 describes the anticipated flow of our framework, and Section 5 presents experimental results, evaluation measures, etc. The findings are presented in the final section, along with a few probable constraints and future developments.

2. Related works

Thorough examination of previously presented approaches to viral genome prediction from DNA is reviewed in this section. Notably, very little work has been done in this area using ML methods; rather, recent works were completely based on deep learning (DL) techniques, and it is impossible to pinpoint the criteria upon which the networks made classification conclusions.

In addition to the alignment-based classification by BLAST mentioned earlier, the algorithm HMMER3 (Mistry et al., 2013) uses profile hidden Markov models (pHMMs) with the vFams database, which was created using multiple sequence alignments from viral proteins in RefSeq (Skewes-Cox et al., 2014). This technique compares sequences to complete viral families, which greatly improves the program's ability to find distant homologs (Bzhalava, Hultin, et al., 2018). However, these processes require a reference database, which is a disadvantage when analyzing highly diverse viral samples.

In (Bzhalava, Tampuu, et al., 2018), utilizing relative synonymous codon usage frequency (RSCU) to assess metagenomic sequencing data, viral sequences were discovered using ML approaches. The authors created a virus/nonvirus classifier using genes from NCBI GenBank. A cross-validation method based on the random forest (RF) algorithm yielded nearly flawless accuracy on this dataset. The frameworks developed with data from NCBI GenBank, on the other hand, performed poorly in regard to classifying contigs acquired from metagenomics studies. They used a metagenomic sequencing dataset produced by the application of NGS technologies to human biospecimens to train the RF algorithm and neural networks (NN) as the main contribution. The algorithms had an area under the receiver operating characteristic (ROC) curve (AUC) of 0.79 and accuracies substantially beyond chance. The discriminative potential of 2 codons (TCG and CGC) was discovered to be exceptionally high.

In (Amgarten et al., 2018), MARVEL was introduced as a method for predicting bacteriophage sequences in metagenomic bins. MARVEL is based on an RF ML technique. To train the software and test it, the authors employed a database containing 1,247 phage and 1,029 bacterial genomes. They showed that distinguishing bacterial from phage sequences with high accuracy only required the extraction of gene density, strand shifts, and the percentage of relevant hits to the viral protein database.

VirSorter (Roux et al., 2015) builds on PhiSpy's work by combining various types of evidence, i.e.,: the enrichment of viral-like and uncharacterized genes, the presence or absence of viral hallmark genes, Pfam-affiliated gene depletion and the identification of gene-enriched

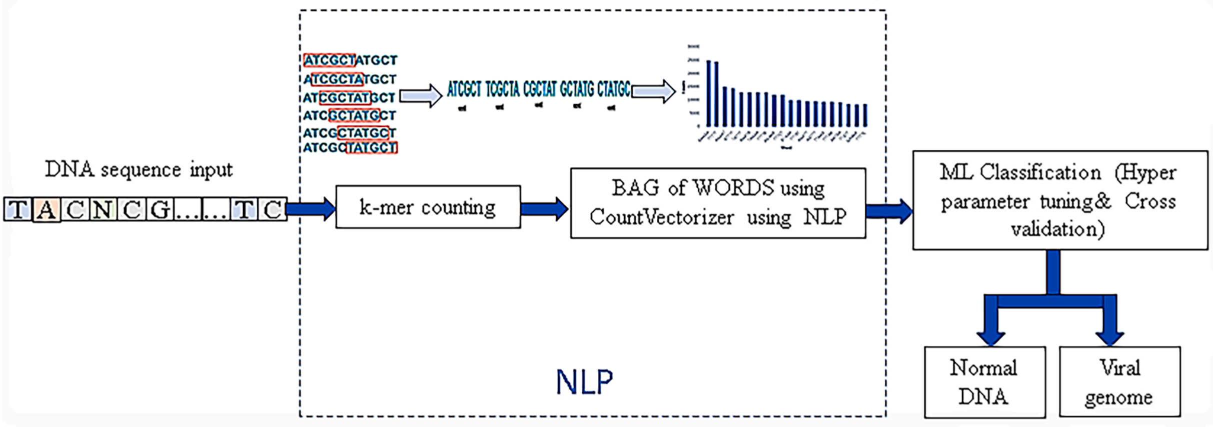


Fig. 1. Workflow outline.

regions with similarity to viral database sequences. Similarity searches against already existing viral databases provide the backbone of VirSorter. The limitations of this classification tool are as follows: VirSorter is a gene-based algorithm that requires a minimum of three anticipated genes within a contig for prediction, which eliminates shorter contigs. A signature viral structural gene is also required for high-confidence VirSorter predictions, which again restricts fragmented viral contig detection. Additionally, VirSorter might overlook numerous novel viruses whose distinguishing genes have yet to be discovered or are underrepresented in reference databases as well as novel viruses that are poorly represented in the current viromes. VirFinder (Ren et al., 2017) uses k-mer frequency instead of gene-based similarity searches in its virus contig identification ML framework. VirFinder exploits the empirical discovery that viruses and hosts have distinct k-mer fingerprints to identify viral sequences. VirFinder performs better than VirSorter. Most of the limitations associated with VirSorter were eliminated in VirFinder.

Notably, none of the ML frameworks described above for finding viruses in metagenomic data were developed or validated to detect viruses in human biospecimens. DL was used to develop DeepVirFinder (Ren et al., 2020), a reference and alignment-free ML technique for finding viral sequences in metagenomic data. When trained on viral RefSeq, DeepVirFinder outperformed VirFinder. The ability to identify viral families that are underrepresented is further enhanced by supplementing the training data from metavirome samples with millions of pure viral sequences. Using DeepVirFinder on actual human gut metagenomic data, 51,138 viral sequences were discovered in people with colorectal cancer that belonged to 175 bins.

In one of the works (Maarala et al., 2018), the authors presented the parallel pipeline ViraPipe, carrying out viral metagenome analysis using Apache Spark on a computing cluster, to address the problem of virus identification from a large number of samples quickly using available techniques. In a flexible and reusable manner, ViraPipe included frequently used genomics tools, MegaHit De Novo assembler, BWA aligner, HMMER3 and BLAST search tools.

To discover viral sequences from metagenomes, RNN-VirSeeker, a DL technique (Liu et al., 2022), was developed. It is trained using 500 bp sequences taken from genomes of known viruses and hosts, and it outperformed VirSorter, VirFinder, and DeepVirFinder by exhibiting 92.11 % precision, 0.9175 AUROC and 86.40 % recall.

ViraMiner (Tampuu et al., 2019) is another convolutional neural network (CNN)-based technique that accepts raw DNA sequences and outputs the probability that the input sequence is viral. Convolutional layers with learnable filters were used to examine the raw sequences. As a result, the model can assess all data and learn which features are critical to extract from labeled instances. To increase the model's capacity for significant information extraction, the ViraMiner model incorporates 2 distinct convolutional branches. How well specific patterns are matched along the DNA sequence is reported by pattern branch. Pattern frequencies are returned by the frequency branch. The output node is connected to these combined branch outputs. In this node, the sigmoid activation function is used to convert the weighted sum of inputs to probability.

Virtifier (Miao et al., 2022), a DL-based viral identifier for metagenomic data sequences, comprises Seq2Vec, a method for encoding meaningful nucleotide sequences, and an attention-based long short-term memory (LSTM) network. Seq2Vec can quickly extract associations by encoding the associations between codons in a nucleotide sequence using a fully trained embedding matrix. The LSTM neural network may further be used to investigate codon connections and sort final features contributing bits when connected with the attention layer.

While existing computational approaches can be used to identify viral genomes, the classification effectiveness is exclusively determined by the structural data recovered. By automatically extracting classification features, deep neural network (DNN) models have exhibited remarkable performance, whereas the degree of model explainability seems very low. Hence, in (Dasari & Bhukya, 2022), CNN-LSTM-based approaches (EdeepVPP, EdeepVPP-hybrid) were developed that extract features automatically. Additionally, EdeepVPP performs model interpretability through trained filters to pinpoint the most crucial patterns associated with viral genomes. EdeepVPP uses feature maps of viral sequences to extract key biological patterns with a programmable CNN model.

The DL model DLmeta (Zhang et al., 2022) was created by the authors for metagenomic identification; it integrates Transformer and CNN to obtain domains through gene and protein domain prediction. They used data from plasmids, bacteria and viruses. In an ablation experiment, they showed that the model can significantly enhance the performance of metagenomic identification and uses CNN for collecting local and Transformer to obtain global characteristics.

In this study, to resolve the abovementioned drawbacks in (Bzhalava, Hultin, et al., 2018; Roux et al., 2015), we created an ML framework for identifying possible viral sequences in raw metagenomic contigs using a combination of NLP and ML algorithms. DL networks, which are very much a "black box" in that researchers still do not completely grasp their "insides," have been utilized in several recent publications. Some of the reviewed works (Roux et al., 2015) needed a large reference database to compare and identify viral genomes in human DNA. However, our model performs efficiently and faster than these approaches without the need for such a reference database. We employed 19 metagenomic studies of different human samples to train the algorithm. We used k-mer counting and the bag-of-words technique utilizing CountVectorizer of NLP to transform the DNA data to a suitable format for ML classification model training. The model can perform with substantially greater accuracy and more quickly than other currently available approaches for viral detection from metagenomic samples. To the best of our knowledge, the suggested framework represents the first precise and fast model that uses an NLP-ML framework for viral detection in raw metagenomic contigs from a variety of human biospecimens and can be modified into a CAD system. Additionally, the model is generalized and not limited to a given viral family. By using these ML algorithms, the proposed model becomes quite easy to interpret and understand.

Our work is closer to (Ren et al., 2017, 2020; Tampuu et al., 2019), where (Ren et al., 2017) is the first ML approach based on k-mers and (Ren et al., 2020) is the first reference-free DL approach. In (Tampuu et al., 2019), one-hot encoded raw sequences were input into a DL architecture (convolutional branches-concatenate-fully connected layer). To develop a simple reference-free automatic framework, in our proposed work, we processed the input DNA sequence by NLP and later applied conventional ML classifiers.

3. Materials AND methods

This section outlines the approaches utilized to generate the binary classification model by fusing NLP and ML algorithms and the database utilized to create the suggested ML model.

3.1. Database

NGS systems: NextSeq, MiSeq, and HiSeq (Illumina) were used to generate the metagenomic sequences used in this study. Human samples from various patient groups were used to create the dataset. The purpose of these investigations was to find viral genomes in ill or healthy people. A benchmarked bioinformatics workflow was used to process and analyze all of the sequencing experiments. PCJ-BLAST was used to examine and label 19 distinct NGS experiments after applying de novo genome assembly algorithms to the training dataset. The following were the PCJ-BLAST criteria: nucleotide match reward = 1; cost to open a gap = 0; type of algorithm = BLASTN; nucleotide mismatch penalty = 1; e-value $\leq$e−4; cost to expand a gap = 2. This bioinformatics workflow labeled all assembled contigs, which were then blended to train different ML algorithms. For model construction, labeled sequences were split into equal segments measuring 300 and 500 bp. Each segment was labeled with the name of the original sequence, and the rest of the nucleotides at the end of the contigs were ignored in future examination (Tampuu et al., 2019).

3.2. K-mer counting and bag-of-words NLP approaches

A significant volume of text data can be processed, analyzed, and understood using NLP. NLP tasks break language down into more manageable, fundamental components, investigate relationships between the pieces, and identify how the pieces function together. We can no longer interpret the text using the traditional method because of the vast amounts of text data and the extremely unstructured data source. This is where NLP comes in. The exploratory strategy we offer in this research tries to combine the finest NLP and ML classification techniques. Text analysis is first performed on them by an NLP chain, and then classification techniques are used to help the model be effectively trained.

To modify the data to be suitable as an ML classification training input, we implemented k-mer counting and bag-of-words approaches from NLP. K-mers (Solis-Reyes et al., 2018) represent k-length substrings within the biological sequence. K-mers, formed of nucleotides, are mostly used in the context of computational genomics and sequence analysis. The acronym AGAT has four monomers (A, G, A, and T), three 2-mers (AG, GA, AT), two 3-mers (AGA and GAT), and one 4-mer. Typically, “k-mer” refers to all of a sequence’s subsequences having length k (AGAT). Many bioinformatics techniques depend on counting k-mers. Although it seems straightforward in theory, counting k-mers in sizable contemporary sequence datasets can quickly exceed the memory capacity of typical computers.

We used the bag-of-words method to convert variable-length texts into fixed-length vectors since ML algorithms work best with structured, well-specified fixed-length inputs (Juluru et al., 2021). Text modeling with the bag-of-words technique is an NLP approach and a statistical language model, and it is simple and flexible to implement. The moniker “bag of words” comes from the fact that it depicts a sentence as a collection of terms. We can call it the feature extraction method from text data in technical terms. This feature extraction technique from documents is simple and adaptable. Textual illustration of how frequently words appear in a document refers to a bag of words. Here, we neglect grammatical nuance, word choice, order, structure, etc., and we keep track of word counts, hence the name bag of words. The model considers whether recognized terms are used in the text, not where exactly they are used. Through this approach, features will be unique words, and word counts will become feature values. Six-mers in the sequence ACGAGGTACGA are generated, and bags of words are created, as shown in Fig. 2.

4. Proposed methodology

In this section, we present an ML classification model that can predict the presence of viral contigs from the coding sequences of human DNA. The complete workflow of the proposed framework is shown in Fig. 3, where it inputs raw DNA sequences and outputs binary classification of DNA with viral genomes. Building and incorporating such an automatic framework into CAD systems will aid in the easy identification of DNA with viral genomes and promote timely treatment by efficiently using healthcare resources.

The suggested framework’s goal in this case is to detect the presence of any viral sequence in human DNA, even if the virus was just recently discovered and the sequence is not yet in the reference database. Pattern observation could not complete the suggested work effectively because it already performs similar activities by comparing it to templates that have been stored. Hence, the entire framework consists of two phases and is developed by combining NLP and ML. First, the input raw DNA sequence is fed into the NLP phase. In this section, k-mer counting and the bag-of-words method using a CountVectorizer are implemented. Therefore, the lengthy raw DNA sequence is divided into fixed length words and later converted to vectors. Then, using these vectors, ML conventional classifiers are trained and fine-tuned to obtain the final optimum model.

Assembled metagenomic contigs built from 19 independent experiments sequenced from a variety of human samples (skin, condylomata, serum, etc.) were used to train the suggested models. Eighty percent training and 10 % testing sets were created by combining, splitting, and rearranging contigs. None of the aforementioned approaches provides the consistent length vectors needed for classifier input, which remains a hurdle. To produce vectors of consistent length using the above approaches, we must resort to truncating sequences or padding with “n” or “0.” The language of life can be understood metaphorically as DNA and protein sequences; it encodes both instructions and functions for the molecules found in all living things. Genome is similar to a book, with k-mers and peptides standing in for words, nucleotide bases and amino acids for the alphabet, and genes and families for sentences and chapters. Given the similarities, it makes sense that groundbreaking NLP research would also apply to the natural language of DNA and protein sequences.

As a first step, we initially divided the lengthy biological sequence into k-mer-length overlapping “words”. For example, the sequence “ATGCATGCA” becomes ’ATGCAT’, ’TGCATG’, ’GCATGC’, and

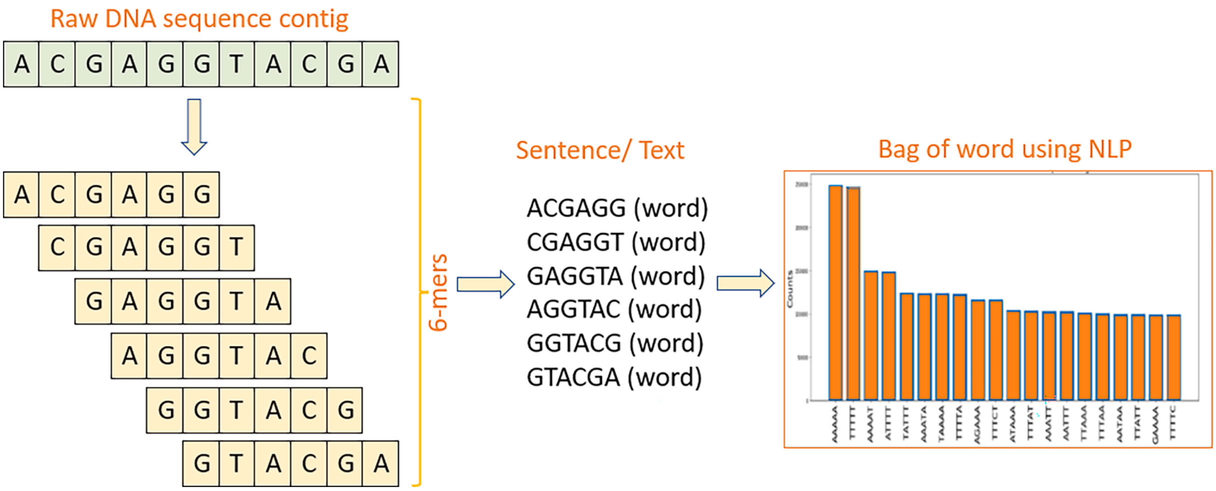


Fig. 2. Six-mers in the sequence ACGAGGTACGA and bag-of-words formation.

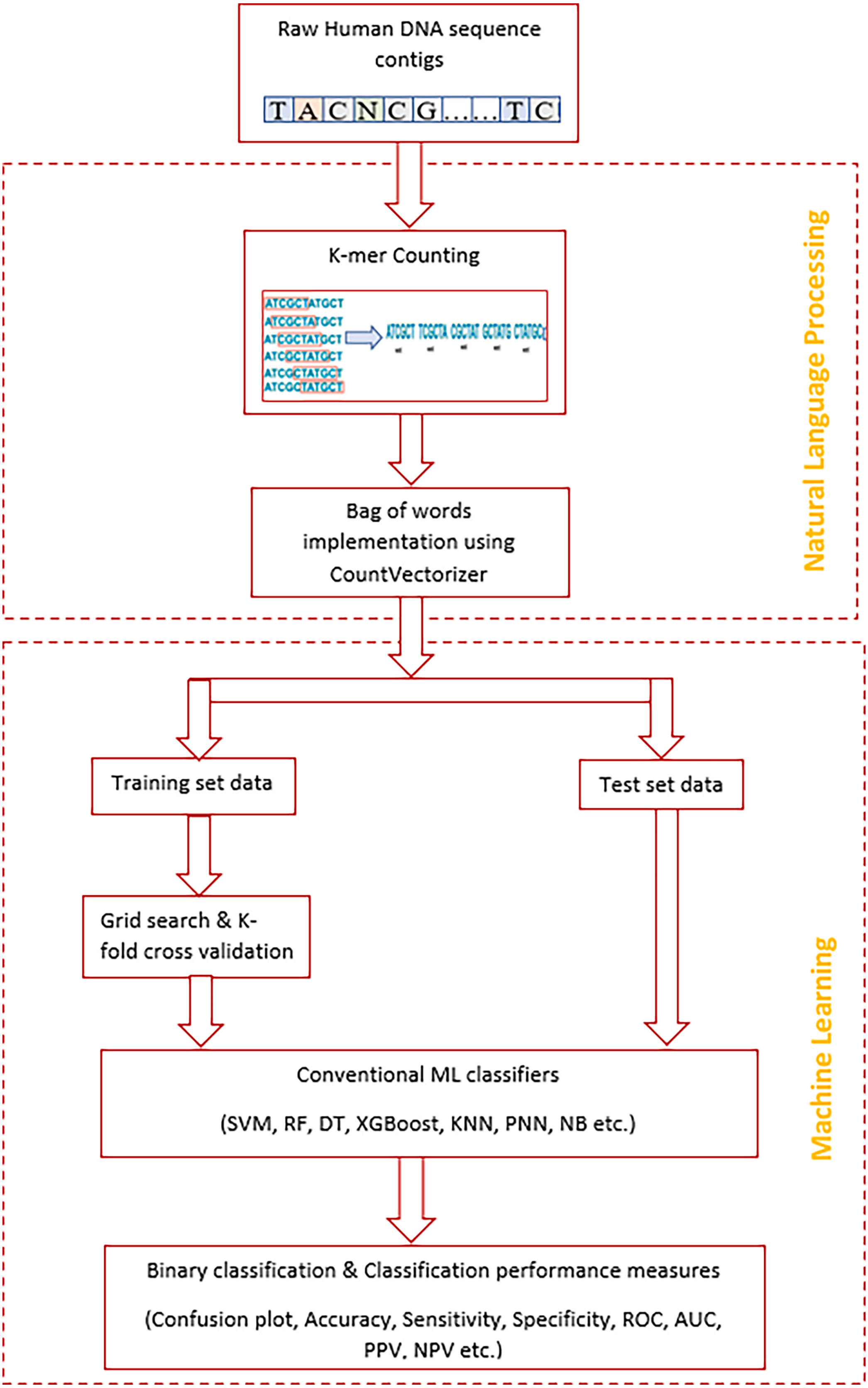

Raw Human DNA sequence contigs
TACNCG……TC
Natural Language Processing
K-mer Counting
ATCGCTATGCT
ATCGCTATGCT
ATCGCTATGCT
ATCGCTATGCT
ATCGCTATGCT
ATCGCTATGCT
ATCGCT TCGCTA CGCTAT GCTATG CTATGC
Bag of words implementation using CountVectorizer
Machine Learning
Training set data
Test set data
Grid search & K-fold cross validation
Conventional ML classifiers
(SVM, RF, DT, XGBoost, KNN, PNN, NB etc.)
Binary classification & Classification performance measures
(Confusion plot, Accuracy, Sensitivity, Specificity, ROC, AUC, PPV, NPV etc.)

Fig. 3. Complete methodology framework.

'CATGCA' if "words" of length 6 hexamers are used. As a result, our sample sequence is divided into four hexamers. Hexamer "words" are being used here; however, this is simply for convenience, and word length can be altered to fit the circumstance. The word length and the degree of overlap must be decided empirically for the specific application. These types of manipulations are referred to in genomics as k-mer counting. Although specialized tools are available to perform this task, Python's NLP tools are extremely simple to use. We employed a user-defined Python function to create k-mers. Then, we created short, overlapping k-mers with a length of 6 using the training data sequences. To be ready for the following step, our coding sequence data were transformed to lowercase and divided into all potential 6-k-mer words.

K-mer counting was performed by scikit-learn NLP tools; thus, the lists of k-mers for each gene were translated into string sentences of words that CountVectorizer could use. Class labels were placed into a new variable called 'y'. Then, we used CountVectorizer and NLP to apply the bag-of-words method to perform feature extraction from the abovementioned texts for modeling ML algorithms. By this method, texts are converted into fixed length vectors. Machines cannot understand letters or words. As a result, while working with text data, they must be expressed mathematically for the machine to comprehend it. Here, the tool for converting text to numbers was CountVectorizer. Each unique word's word count is shown by a row. Text data can be used directly in ML and DL models with CountVectorizer.

These features were then fed into typical conventional classification algorithms, which allowed multiple ML classification models' classification performance metrics to be compared. Classifiers such as random forest (RF), decision tree (DT), XGBoost, SVM, KNN and naive Bayes (NB) were employed and fine-tuned using GS and tenfold cross-validation. We preferred the GS approach because it is simple to use and understand and is robust in its ability to predict outcomes. With this tuning method, we created a model and assessed it for each possible combination of the various hyperparameters. All of the classification algorithms we used have a possible range of values for each parameter. Here, the gridsearch technique was used to find the best hyperparameters using 'GridSearchCV' from sklearn.

We used classical approaches for classification because they do not require as much computer power, allow for faster iterations, and allow for the rapid testing of numerous methodologies. These algorithms are simple to perceive and comprehend because conventional ML involves direct feature engineering. Additionally, because we have a deeper grasp of the data and underlying algorithms, modifying hyperparameters and changing model designs is simpler. Deep networks are very much a "black box" in that academics still do not fully comprehend their "insides". Each ML trained model exports a variety of classification evaluation metrics, including the f1 score, accuracy, recall and precision.

5. Experimental results and discussion

The experimental findings of each phase of the proposed model will be discussed here. The suggested models were trained using metagenomic contigs that were constructed from 19 independent studies and sequenced from a variety of human samples. First, the model received raw DNA data as input, as shown in Fig. 4.

At this point, we have a large list of sequences, and we are unaware of how many vectors we have to consider. We cannot feed these sequences directly into the model. We can, however, convert the DNA sequences into languages by using k-mer counting. Therefore, we used k-mer counting to make a fixed set of count variables. From all the sequences, the k-mer value was finalized empirically, as shown in Fig. 5. Generated 6-mers appeared as illustrated in Fig. 6. Basically, we created fixed vectors, i.e., vectors of uniform length. These words were converted into vectors using NLP after first converting the data to lowercase; otherwise, they may be considered separate lists in some cases.

Then, all sequences were combined, and it became easy to convert them into bags of words. All text was converted into lists, and the list values were combined with a blank space in between. A sample of such a conversion is shown in Fig. 7.

Consequently, independent features in the form of strings were obtained. In NLP, we cannot directly use these strings in a model. Therefore, we converted these strings into bags of words using CountVectorizer. A sample bag-of-words diagram is shown in Fig. 8 since the complete diagram cannot be accommodated within a single figure window. To represent the class variables, normal human DNA sequences were labeled 'zero', and DNA with the presence of a viral genome was labeled 'one' to prepare the data to be ready for ML model training. Contigs were concatenated, shuffled, and divided into 80 % for training and 20 % for testing. The shape of the features fed into the model is (169859, 262144) for the training set and (42465, 262144) for the test set.

6. Classification performance measures

The following metrics were used to gauge how well our classification model performed (Alshayeji et al., 2021).

True Positive (TP): Human DNA contig having viral genome, properly predicted by the classifier.

False Positive (FP): Human DNA contigs that do not have a viral genome but were recognized as belonging to the class of viral genomes by the classifier.

True Negative (TN): Human DNA contig that does not belong to the viral genome class, properly predicted by the classifier as normal.

False Negative (FN): Human DNA contig, which belongs to the viral genome class but is predicted to be normal by the classifier.

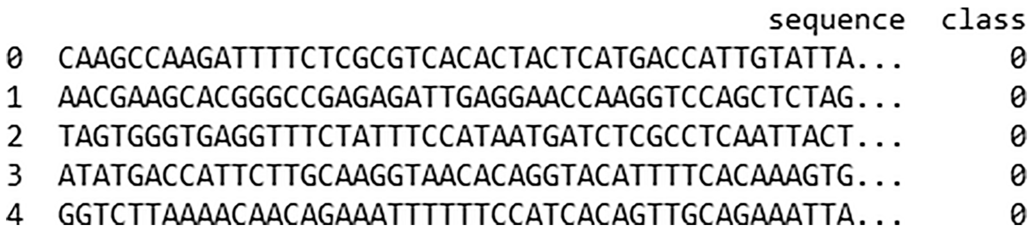
Accuracy: This measurement shows how accurately the model distinguishes between DNA contigs with normal and viral genome instances (Eq. (1)):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Recall or TPR or sensitivity: This measure shows the proportion of viral genome-containing human DNA contigs that the classifier properly detected (Eq. (2)).

$$Recall\ or\ Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

Precision or Positive Predictive Value (PPV): By comparing the correct true positives to the predicted ones, it determines how accurately the model performs (Eq. (3)).



```
                                            sequence  class
0  CAAGCCAAGATTTTCTCGCGTCACACTACTCATGACCATTGTATTA...      0
1  AACGAAGCACGGGCCGAGAGATTGAGGAACCAAGGTCCAGCTCTAG...      0
2  TAGTGGGTGAGGTTTCTATTTCCATAATGATCTCGCCTCAATTACT...      0
3  ATATGACCATTCTTGCAAGGTAACACAGGTACATTTTCACAAAGTG...      0
4  GGTCTTAAAACAACAGAAATTTTTTCCATCACAGTTGCAGAAATTA...      0
```

Fig. 4. Input raw DNA data sample.

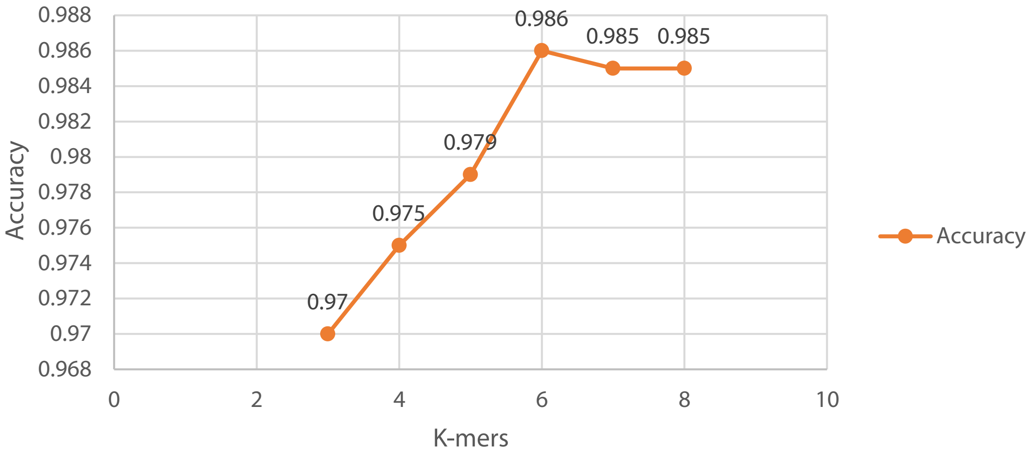
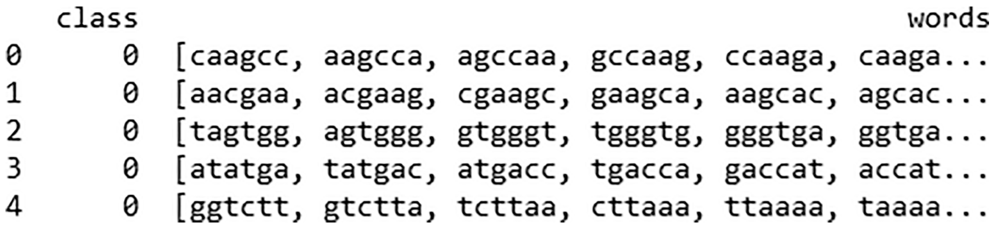
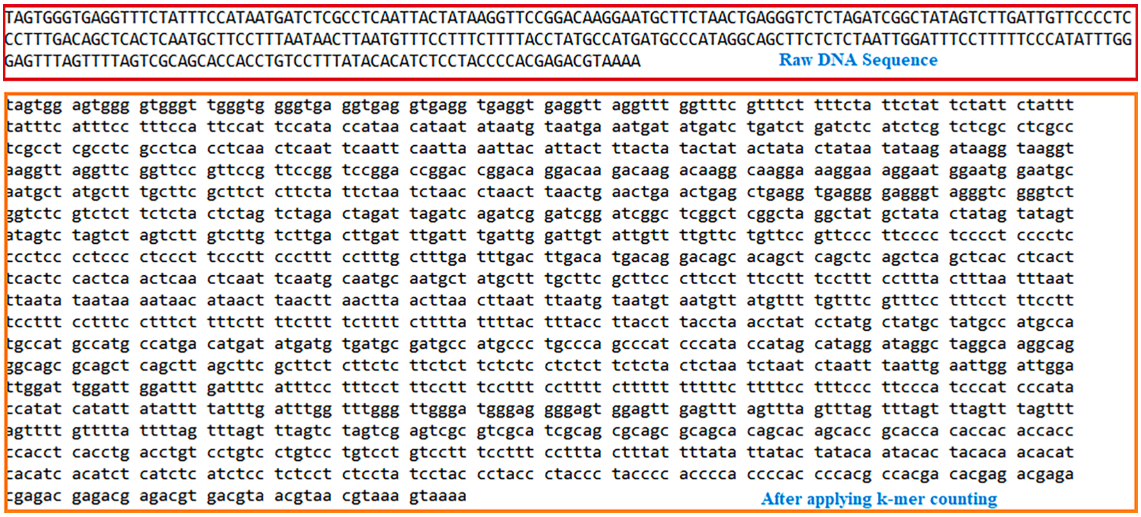


Fig. 5. K-mers vs accuracy plot.



```
   class                                          words
0      0  [caagcc, aagcca, agccaa, gccaag, ccaaga, caaga...
1      0  [aacgaa, acgaag, cgaagc, gaagca, aagcac, agcac...
2      0  [tagtgg, agtggg, gtgggt, tgggtg, gggtga, ggtga...
3      0  [atatga, tatgac, atgacc, tgacca, gaccat, accat...
4      0  [ggtctt, gtctta, tcttaa, cttaaa, ttaaaa, taaaa...
```

Fig. 6. Sample visualization of generated 6-mers.



```
TAGTGGGTGAGGTTTCTATTTCCATAATGATCTCGCCTCAATTACTATAAGGTTCCGGACAAGGAATGCTTCTAACTGAGGGTCTCTAGATCGGCTATAGTCTTGATTGTTCCCCTC
CCTTTGACAGCTCACTCAATGCTTCCTTTAATAACTTAATGTTTCCTTTCTTTTACCTATGCCATGATGCCCATAGGCAGCTTCTCTCTAATTGGATTTCCTTTTTCCCATATTTGG
GAGTTTAGTTTTAGTCGCAGCACCACCTGTCCTTTATACACATCTCCTACCCCACGAGACGTAAAA                Raw DNA Sequence
```

```
tagtgg agtggg gtgggt tgggtg gggtga ggtgag gtgagg tgaggt gaggtt aggttt ggtttc gtttct tttcta ttctat tctatt ctattt
tatttc atttcc tttcca ttccat tccata ccataa cataat ataatg taatga aatgat atgatc tgatct gatctc atctcg tctcgc ctcgcc
tcgcct cgcctc gcctca cctcaa ctcaat tcaatt caatta aattac attact ttacta tactat actata ctataa tataag ataagg taaggt
aaggtt aggttc ggttcc gttccg ttccgg tccgga ccggac cggaca ggacaa gacaag acaagg caagga aaggaa aggaat ggaatg gaatgc
aatgct atgctt tgcttc gcttct cttcta ttctaa tctaac ctaact taactg aactga actgag ctgagg tgaggg gagggt agggtc gggtct
ggtctc gtctct tctcta ctctag tctaga ctagat tagatc agatcg gatcgg atcggc tcggct cggcta ggctat gctata ctatag tatagt
atagtc tagtct agtctt gtcttg tcttga cttgat ttgatt tgattg gattgt attgtt ttgttc tgttcc gttccc ttcccc tcccct cccctc
ccctcc cctccc ctccct tccctt cccttt cctttg ctttga tttgac ttgaca tgacag gacagc acagct cagctc agctca gctcac ctcact
tcactc cactca actcaa ctcaat tcaatg caatgc aatgct atgctt tgcttc gcttcc cttcct ttcctt tccttt ccttta ctttaa tttaat
ttaata taataa aataac ataact taactt aactta acttaa cttaat ttaatg taatgt aatgtt atgttt tgtttc gtttcc tttcct ttcctt
tccttt cctttc ctttct tttctt ttcttt tctttt ctttta ttttac tttacc ttacct taccta acctat cctatg ctatgc tatgcc atgcca
tgccat gccatg ccatga catgat atgatg tgatgc gatgcc atgccc tgccca gcccat cccata ccatag catagg ataggc taggca aggcag
ggcagc gcagct cagctt agcttc gcttct cttctc ttctct tctctc ctctct tctcta ctctaa tctaat ctaatt taattg aattgg attgga
ttggat tggatt ggattt gatttc atttcc tttcct ttcctt tccttt cctttt cttttt tttttc ttttcc tttccc ttccca tcccat cccata
ccatat catatt atattt tatttg atttgg tttggg ttggga tgggag gggagt ggagtt gagttt agttta gtttag tttagt ttagtt tagttt
agtttt gtttta ttttag tttagt ttagtc tagtcg agtcgc gtcgca tcgcag cgcagc gcagca cagcac agcacc gcacca caccac accacc
ccacct cacctg acctgt cctgtc ctgtcc tgtcct gtcctt tccttt ccttta ctttat tttata ttatac tataca atacac tacaca acacat
cacatc acatct catctc atctcc tctcct ctccta tcctac cctacc ctaccc tacccc acccca ccccac cccacg ccacga cacgag acgaga
cgagac gagacg agacgt gacgta acgtaa cgtaaa gtaaaa                              After applying k-mer counting
```

Fig. 7. Sample of created lists to apply the bag-of-words technique.

$$Precision\ or\ PPV = \frac{TP}{TP + FP} \tag{3}$$

F1-Scores: Harmonic mean of precision and recall conveys the balance between these 2 measures (Eq. (4)).

$$F1\text{-}score = 2*\frac{Precision*Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

Table 1 shows the classification performance metrics for the DT, KNN, RF, XGBoost, NB and SVM classifiers. Additionally, the Matthews correlation coefficient, kappa, classification success index and detection rate, etc., are given in Table 2 for the same set of ML models. The KNN classifier produced the best results with a classification accuracy of 98.6 % to precisely classify DNA sequences with the presence of the viral genome from normal human DNA. The Matthews correlation coefficient of 0.896 and Kappa value of 0.895 specify that the developed model gives better predictions. Both of the measures give similar results. The Cohen kappa score measures how well the proposed ML classifier prediction matches the ground truths or how well it performs compared to a random classifier. The classification success index is another evaluation measure that exclusively focuses on the positive class. The detection rate signifies the correct predictions made by the trained model while considering the overall data. All these measures clearly evaluate an ML model, especially the models trained with imbalanced datasets.

While comparing the evaluation performances of these classifiers, all the classifiers were capable of predicting the viral genomes in the input human DNA. However, the KNN classifier had the best results, i.e., 98.6

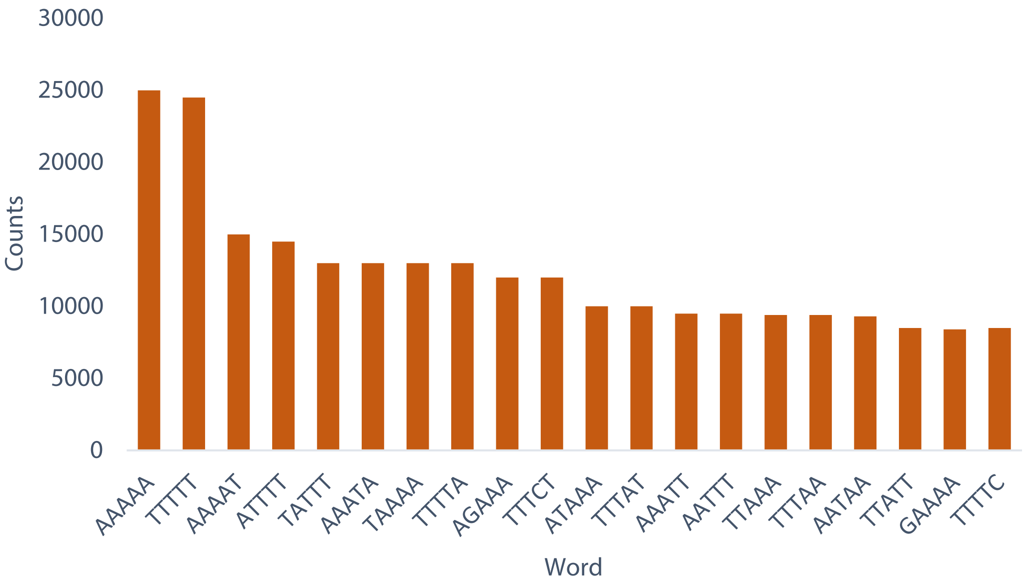


Fig. 8. Bag-of-words diagram.

Table 1
Classification performance measures of ML conventional algorithms.

Classifier	Accuracy	Precision/PPV	Recall/TPR/Sensitivity	F1-score	Specificity/TNR	NPV
XGBoost	0.980	0.981	0.980	0.972	0.99	0.97
RF	0.981	0.981	0.981	0.974	0.99	0.9812
DT	0.980	0.976	0.980	0.970	0.99	0.9797
KNN	0.986	0.985	0.986	0.984	1	0.9879
SVM	0.981	0.981	0.981	0.973	0.99	0.981
NB	0.984	0.981	0.984	0.980	0.9854	0.98

Table 2
Classification performance measures of ML conventional algorithms.

Classifier	Matthews Correlation Coefficient	Kappa	Classification Success Index	Detection Rate
XGBoost	0.639	0.590	0.961	0.980
RF	0.696	0.654	0.962	0.981
DT	0.648	0.595	0.956	0.980
KNN	0.896	0.895	0.971	0.986
SVM	0.696	0.654	0.962	0.981
NB	0.712	0.710	0.965	0.984

% classification accuracy along with 98.5 % precision, 98.6 % recall, 0.984 F1 score, 0.896 Matthews correlation coefficient, 0.895 kappa, 0.97 classification success index, etc. From these results, it is evident that the suggested model can accurately predict raw DNA sequences with viral genome presence and those without it. The normalized confusion matrix plot obtained for the finalized KNN classifier is shown in Fig. 9.

The ROC curve and precision-recall curve obtained from the logistic regression and no skill classifier are given in Fig. 10 and Fig. 11. Lower FPs and larger TNs are shown by lower values on the x-axis of the plot. Higher TPs and fewer FNs are shown by the plot's y-axis values, which are larger. Curves that bend up to the plot's top left are used to illustrate skillful models. A no-skill classifier cannot distinguish classes and consistently predicts random or constant classes. The diagonal line from the bottom to top right of the plot represents the model without any skill at each threshold and has an AUC of 0.5. A line running from the bottom left of the plot to the top left and then across the top to the top right represents a model with perfect skill. A precision-recall curve is then made, comparing a logistic regression (orange) and no skill (blue) model for precision/recall for each threshold.

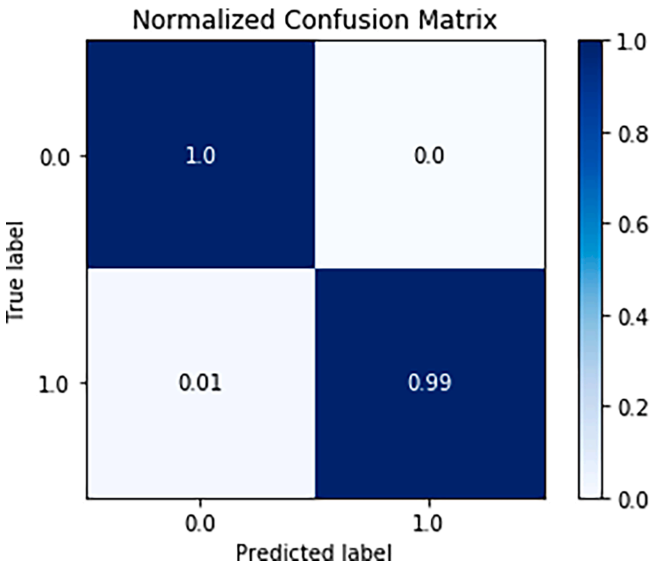


Fig. 9. Normalized confusion plot for the KNN classifier.

A comparison of the suggested work with some of the state-of-the-art models is given in Table 3, and from this, it is evident that the proposed work is simpler and performs better than the other models.

7. Conclusion

To identify viral genomes in raw human DNA contigs, we created and

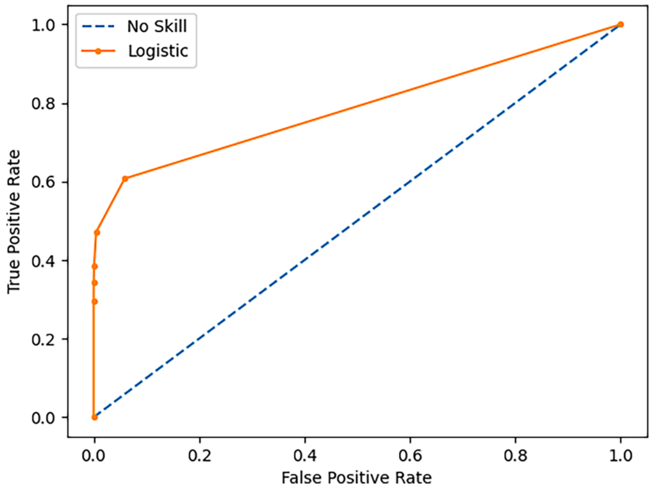


Fig. 10. ROC curve.

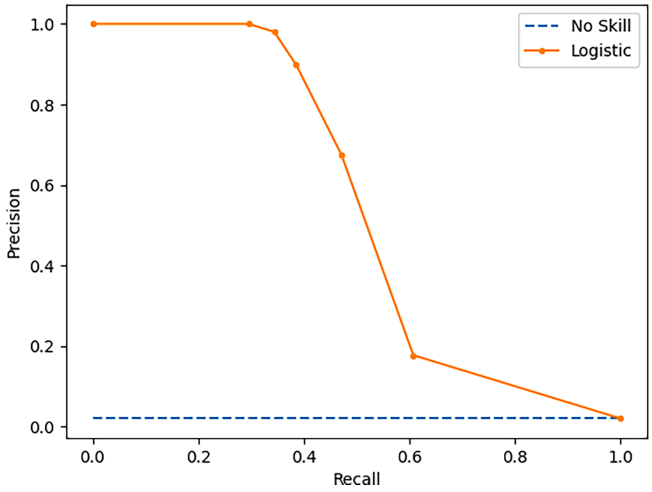


Fig. 11. Precision-recall curve.

Table 3
Comparison table.

Reference	Method	Results
(Tampuu et al., 2019)	ViraMiner (CNN framework)	98 % precision, 98 % recall, AUROC score (0.923)
(Chaudhary et al., 2015)	Short 16S rRNA HVRs taxonomic classification with 16S Classifier using RF	91 % precision, 99.0 % accuracy, 98 % sensitivity
(Vervier et al., 2016)	Rank-flexible machine learning-based compositional approach	97.5 % precision
(Bzhalava, Tampuu, et al., 2018)	Trained RF and NN on metagenomic sequencing dataset generated by NGS technologies	97 % precision, 97 % recall, AUROC score (0.996)
(Ren et al., 2020)	End to end DL network: DeepVirFinder	AUROC score (0.98)
(Liu et al., 2022)	DL network: RNN-VirSeeker	92.11 % precision, 86.4 % recall, AUROC (0.9175)
(Miao et al., 2022)	Encoding with Seq2Vecn and classification by attention-based LSTM network	91.14 % precision, 92.6 % recall,0.9186 F1-score
(Dasari & Bhukya, 2022)	CNN – LSTM	AUROC: 0.98 (EdeepVPP), 0.99 (EdeepVPP-Hybrid)
Proposed work	Human raw DNA input-NLP (k-mer counting, bag of words)-MLP framework	98.6 % accuracy, 98.5 % precision, 98.6 % sensitivity, 98.54 % specificity

implemented an automatic model by combining NLP and ML techniques. In general, human samples contain a variety of viruses, and their structure varies from one sample to another. Human viral detection and classification will continue to be a serious challenge in the future. Every day, investigations and tests generate massive amounts of biological evidence, high-dimensional data and massive data sizes. Many people are unaware that viruses change or mutate, similar to any other living organism attempting to survive. Hence, an automatic viral genome prediction model from raw DNA is essential.

Sequences from 19 metagenomic investigations were used to create the proposed model, which was examined and classified using BLAST. Raw human DNA contigs were converted into a suitable form to serve as input into ML algorithms by implementing NLP techniques such as k-mer counting and the bag-of-words technique. The proposed model achieved 98.6 % classification accuracy using the KNN classifier along with 98.5 % precision, 98.6 % recall, 0.984 F1 score, 0.896 Matthews correlation coefficient, 0.895 kappa, 0.97 classification success index and detection rate of 98.6 % for the binary classification of DNA with viral genomes. The proposed framework can accurately predict viral sequences from novel data, meaning that the model can accurately anticipate even unknown viral sequences and hence function as a recommendation system. Once another large labeled metagenomics database of human samples is available, more experiments will be conducted to enhance and prove the proposed model superiority as a recommendation system. The suggested models are also universal and not specific to a particular viral family. By implementing this into the CAD system, day-to-day hospital activities become easier since our model is precise and fast. Health care professional resources could be effectively used even under the pandemic situation. In the future, we plan to further develop this work by implementing a DL model that can identify or visualize the exact viral class of the genome rather than providing binary classification. Therefore, early predictions could be achieved by automatic artificial intelligence (AI) models, and mutation of viruses could be limited to some extent.

CRediT authorship contribution statement

Mohammad H. Alshayeji: Conceptualization, Methodology, Formal analysis, Validation, Supervision, Writing – original draft, Writing – review & editing. **Silpa ChandraBhasi Sindhu:** Software, Investigation, Writing – original draft, Visualization. **Sa'ed Abed:** Formal analysis, Validation, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This research has received no external funding.

References

Alshayeji, M., Al-Buloushi, J., Ashkanani, A., & Abed, S. (2021). Enhanced brain tumor classification using an optimized multi-layered convolutional neural network architecture. *Multimedia Tools and Applications, 80*(19), 28897–28917. https://doi.org/10.1007/s11042-021-10927-8

Amgarten, D., Braga, L. P. P., da Silva, A. M., & Setubal, J. C. (2018). MARVEL, a tool for prediction of bacteriophage sequences in metagenomic bins. *Frontiers in Genetics, 9* (AUG), 304. https://doi.org/10.3389/FGENE.2018.00304/BIBTEX

BLAST: Basic Local Alignment Search Tool. (n.d.). Retrieved April 21, 2022, from https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD%20=%20Web&PAGE_TYPE%20=%20BlastDocs&DOC_TYPE%20=%20DeveloperInfo.

Bzhalava, Z., Hultin, E., & Dillner, J. (2018). Extension of the viral ecology in humans using viral profile hidden Markov models. *PLoS ONE, 13*(1), e0190938.

Bzhalava, Z., Tampuu, A., Bała, P., Vicente, R., & Dillner, J. (2018). Machine Learning for detection of viral sequences in human metagenomic datasets. *BMC Bioinformatics, 19* (1), 1–11. https://doi.org/10.1186/S12859-018-2340-X/TABLES/2

Chaudhary, N., Sharma, A. K., Agarwal, P., Gupta, A., & Sharma, V. K. (2015). 16S Classifier: A tool for fast and accurate taxonomic classification of 16S rRNA hypervariable regions in metagenomic datasets. *PLoS ONE, 10*(2), e0116106.

Dasari, C. M., & Bhukya, R. (2022). Explainable deep neural networks for novel viral genome prediction. *Applied Intelligence, 52*(3), 3002–3017. https://doi.org/10.1007/S10489-021-02572-3/FIGURES/8

Juluru, K., Shih, H. H., Murthy, K. N. K., & Elnajjar, P. (2021). Bag-of-words technique in natural language processing: A primer for radiologists. *Radiographics, 41*(5), 1420–1426. https://doi.org/10.1148/RG.2021210025/ASSET/IMAGES/LARGE/RG.2021210025.VA.JPEG

Liang, G., & Bushman, F. D. (2021). The human virome: Assembly, composition and host interactions. *Nature Reviews Microbiology 2021 19:8*, *19*(8), 514–527. https://doi.org/10.1038/s41579-021-00536-5.

Liu, F., Miao, Y., Liu, Y., & Hou, T. (2022). RNN-VirSeeker: A deep learning method for identification of short viral sequences from metagenomes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19*(03), 1840–1849. https://doi.org/10.1109/TCBB.2020.3044575

Maarala, A. I., Bzhalava, Z., Dillner, J., Heljanko, K., & Bzhalava, D. (2018). ViraPipe: Scalable parallel pipeline for viral metagenome analysis from next generation sequencing reads. *Bioinformatics (Oxford, England), 34*(6), 928–935. https://doi.org/10.1093/BIOINFORMATICS/BTX702

Meiring, T. L., Salimo, A. T., Coetzee, B., Maree, H. J., Moodley, J., Hitzeroth, I. I., … Williamson, A. L. (2012). Next-generation sequencing of cervical DNA detects human papillomavirus types not detected by commercial kits. *Virology Journal, 9*(1), 1–10. https://doi.org/10.1186/1743-422X-9-164/FIGURES/3

Miao, Y., Liu, F., Hou, T., & Liu, Y. (2022). Virtifier: A deep learning-based identifier for viral sequences from metagenomes. *Bioinformatics, 38*(5), 1216–1222. https://doi.org/10.1093/BIOINFORMATICS/BTAB845

Mistry, J., Finn, R. D., Eddy, S. R., Bateman, A., & Punta, M. (2013). Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions. *Nucleic Acids Research, 41*(12), e121–e. https://doi.org/10.1093/NAR/GKT263

Pinu, F. R., Beale, D. J., Paten, A. M., Kouremenos, K., Swarup, S., Schirra, H. J., & Wishart, D. (2019). Systems biology and multi-omics integration: Viewpoints from the metabolomics research community. *Metabolites, 9*(4). https://doi.org/10.3390/METABO9040076

Ren, J., Ahlgren, N. A., Lu, Y. Y., Fuhrman, J. A., & Sun, F. (2017). VirFinder: A novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome, 5*(1), 69. https://doi.org/10.1186/S40168-017-0283-5/TABLES/2

Ren, J., Song, K., Deng, C., Ahlgren, N. A., Fuhrman, J. A., Li, Y., Xie, X., Poplin, R., & Sun, F. (2020). Identifying viruses from metagenomic data using deep learning. *Quantitative Biology 2020 8:1*, *8*(1), 64–77. https://doi.org/10.1007/S40484-019-0187-4.

Roux, S., Enault, F., Hurwitz, B. L., & Sullivan, M. B. (2015). VirSorter: Mining viral signal from microbial genomic data. *PeerJ, 3*(5). https://doi.org/10.7717/PEERJ.985

Skewes-Cox, P., Sharpton, T. J., Pollard, K. S., & DeRisi, J. L. (2014). Profile hidden Markov models for the detection of viruses within metagenomic sequence data. *PLoS ONE, 9*(8), e105067. https://doi.org/10.1371/JOURNAL.PONE.0105067

Solis-Reyes, S. I., Avino, M., Poon, A., & Kari, L. (2018). *An open-source k-mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes*. https://doi.org/10.1371/journal.pone.0206409.

Tampuu, A., Bzhalava, Z., Dillner, J., & Vicente, R. (2019). ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples. *PLoS ONE, 14* (9), e0222271. https://doi.org/10.1371/JOURNAL.PONE.0222271

Vervier, K., Mahé, P., Tournoud, M., Veyrieras, J. B., & Vert, J. P. (2016). Large-scale machine learning for metagenomics sequence classification. *Bioinformatics, 32*(7), 1023–1032. https://doi.org/10.1093/BIOINFORMATICS/BTV683

Zhang, Y., Li, C., Feng, H., & Zhu, D. (2022). DLmeta: A deep learning method for metagenomic identification. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2022*, 303–308. https://doi.org/10.1109/BIBM55620.2022.9995231