Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 8 (2026) 3606

Journal of the Nigerian Society of Physical Sciences

An explainable CNN–BiLSTM–Random Forest framework for epidemic warning-signal and disease-status classification

Faruk Obansa Muhammed*, Muhammad Aliyu Suleiman, Saleh El-Yakub Abdullahi, Austin Olom Ogar

Department of Computer Science, Nile University of Nigeria, Abuja, Nigeria

Abstract

This study evaluates an explainable ensemble in which Convolutional Neural Network and Bidirectional Long Short-Term Memory feature extractors feed a Random Forest classifier, applied under one architectural template to two independent tasks: labelling Ebola-related tweets as symptom-bearing warning signals and labelling structured coronavirus disease 2019 (COVID-19) patient records as positive or negative for severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The tasks use disjoint data on different diseases, are never fused, and model no forecasting horizon. Results come from five-fold stratified cross-validation repeated twice, with preprocessing and decision thresholds fitted inside each training partition and metrics reported per class. On the social media task, the framework attains a positive-class sensitivity of 0.918, precision of 0.975, and F1-score of 0.946, exceeding a term-frequency baseline by 0.024 in F1-score. Two ablations qualify this: raising the tokenizer vocabulary from 100 to 5,000 terms and training the extractors end-to-end account for the entire gain; the recurrent branch and the sentiment feature add nothing thereafter. On the clinical task, no configuration improves on a Random Forest over the laboratory variables alone, but the operating point matters more than the model: reporting at 90% specificity rather than at a probability cut of 0.5 raises sensitivity from 0.310 to 0.711, and to 0.803 at 80% specificity. Discrimination falls to 0.66 in patients without a complete blood count, so the instrument is a post-haematology triage aid rather than a screening tool. Shapley Additive Explanations rank leukocytes and platelets highest but cover only 14% of the model's attribution mass.

DOI: 10.46481/jnsps.2026.3606

Keywords: Epidemic surveillance, Ensemble learning, Explainable artificial intelligence, Text classification, Disease-status classification

Article History:
Received: 14 June 2026
Received in revised form: 15 July 2026
Accepted for publication: 02 September 2026
Available online: 11 September 2026

Communicated by: S. Folorunso

1. Introduction

Early detection of epidemic outbreaks remains critical for reducing disease spread, minimising mortality, and enabling timely public health interventions. When surveillance and response systems are reactive rather than proactive, outbreaks can escalate rapidly, placing significant strain on healthcare infrastructure and increasing societal and economic burdens [1].

Traditional approaches to outbreak prediction primarily rely on mathematical and epidemiological models that describe transmission dynamics and estimate future cases and deaths [2–4]. However, these models often struggle to adapt to external and rapidly changing factors such as behavioural changes, hygiene practices, social distancing, and emerging variants [5, 6]. To address these limitations, researchers increasingly employ machine learning (ML) and deep learning techniques to capture complex patterns and detect shifts in transmission trends [7–9]. For example, Amin *et al.* [10] applied Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine

*Corresponding Author Tel. No.: +234-813-588-5361.
Email address: faruklincoln@gmail.com (Faruk Obansa Muhammed)

(SVM), and Decision Tree (DT) models to social media data for dengue and flu detection. Aleixo *et al.* [11] utilized a gradient boosting decision tree algorithm (CatBoost) with epidemiological, weather, and socio-demographic data to predict dengue cases, while Kirange [12] developed a stacking ensemble using Google Trends queries and historical case counts for earlier outbreak identification.

Other studies have applied traditional ML methods such as KNN, SVM, and RF, decision tree-based models including XGBoost (XGB), LightGBM (LGBM), CatBoost, and DT [13–16], and deep learning approaches such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Multilayer Perceptron (MLP) [14, 16–18]. Beyond single models, a number of studies have combined several learners into ensembles to push predictive performance further [19–23].

Even with this progress, a handful of persistent weaknesses still hold back the practical use of ML-based early warning tools. One recurring problem is weak sensitivity: many models fail to flag a large share of true cases, and the resulting high false-negative rates undermine their reliability precisely when early intervention matters most [7]. Heterogeneous data sources are also rarely handled within a common methodological framework; text-based online signals and structured clinical records are typically studied in separate literatures, with separate architectures and incomparable reporting conventions, even though both are routinely available to the same surveillance unit [7, 24]. Interpretability tends to be neglected as well, with only a small fraction of studies adopting explainable artificial intelligence (AI), which leaves clinicians with little basis on which to trust or act on the predictions [11]. Together, these gaps narrow the real-world value of current forecasting systems in everyday public health practice.

Building the framework on CNN, LSTM, and Random Forest rather than on Transformer- or attention-based designs was a deliberate engineering decision. Transformers are powerful sequence models, but their $O(n^2)$ self-attention cost and their need for large training corpora make them ill-suited to epidemic prediction, where labelled data are scarce and rapid deployment is essential [25]. A CNN, by comparison, captures local patterns at only $O(n{\cdot}k{\cdot}d)$ per layer [26], and an LSTM models sequential dependencies at $O(n{\cdot}d^2)$, both far lighter than full attention. Using a Random Forest as the final classifier adds two further advantages: it is robust to overfitting on modest datasets and yields feature-importance scores that are interpretable in themselves. It also works hand in glove with TreeExplainer, whose Shapley additive explanations (SHAP) computations are tractable for tree-based models in a way that deep-network SHAP is not [27]. The resulting combination gives up little predictive power while gaining considerably in efficiency and transparency, which makes it well suited to the resource-constrained settings where such systems are most needed.

These difficulties point naturally toward an ensemble strategy, in which models with different inductive biases are combined so that their strengths reinforce one another while their individual shortcomings are dampened. LSTMs track the sequential dependencies that characterise transmission dynamics [28, 29], CNNs are adept at extracting nonlinear and local structure from heterogeneous inputs [30], and Random Forests generalise well, having outperformed classical statistical baselines such as autoregressive integrated moving average (ARIMA) on outbreak tasks [31]. Combining the three within one framework therefore allows each to contribute what it does best. Following this reasoning, the present study develops an ensemble that couples CNN, bidirectional long short-term memory (BiLSTM), and RF into a single architectural template, applies that template separately to a text-based and a record-based epidemic surveillance task, and returns interpretable outputs to support timely, actionable public health responses. The tasks addressed are classification tasks: each instance is assigned a label from evidence already contained in that instance. No outbreak timing, incidence, or future trajectory is forecast at any point.

A variety of newer architectures has lately been applied to epidemic forecasting. Temporal Fusion Transformers (TFTs) pair variable-selection networks with multi-head attention to produce strong multi-horizon forecasts on influenza-like-illness data. Spatio-temporal Graph Neural Networks (GNNs) such as the spatio-temporal graph convolutional network (STGCN) and diffusion convolutional recurrent neural network (DCRNN) capture inter-regional spread by exploiting mobility and adjacency graphs, attention-augmented LSTMs refine the vanilla model by weighting the most informative time steps, and hybrid CNN–LSTM designs have shown promise for coronavirus disease 2019 (COVID-19) case prediction.

A common thread, however, is that these methods concentrate on a single modality, either clinical time series or text-based surveillance, and seldom explain their outputs. The approach taken here differs on two of these points: it builds robustness through a CNN + BiLSTM + Random Forest ensemble, and it supplies post-hoc interpretability through SHAP. On the third it makes no claim to advance beyond them: the two modalities considered here are processed by the same template but never jointly fused, and the datasets describe different diseases, so this work is a pair of modality-specific studies sharing an architecture rather than a multimodal fusion method. It is thus better understood as complementing these single-modality deep learning methods than as competing with them, and Section 3 reports a quantitative comparison against several of them.

This work offers four main contributions. It puts forward an explainable ensemble that brings CNN, BiLSTM, and Random Forest models together for the classification of epidemic warning signals and disease status. It shows that a single architectural template transfers across two very different input modalities, unstructured text and structured laboratory records, and reports both under one common evaluation protocol so that the two are directly comparable. It demonstrates empirically that, on the social media task, the design attains higher positive-class sensitivity and F1-score than the ablations evaluated here, and it reports per-class rather than only averaged metrics so that the size of that gain is not obscured by class imbalance. Finally, it embeds SHAP-based explainability so that the model's decisions can be examined and trusted, an especially important property in healthcare applications.

2. Method

This section presents the dataset and methodology employed to achieve the aim of this study. It details the approach taken to develop an ensemble machine learning model for the classification of epidemic warning signals and disease status, applied to two data sources and made interpretable for practical application.

2.1. Datasets

In the field of infectious disease research, the integration of heterogeneous datasets enhances the ability to monitor outbreaks, predict patterns, and assess population-level impact [32, 33]. Two distinct but complementary data types are used in this study: tweet data and clinical data. Tweet data are unstructured [10, 34–36], while clinical data comprise structured, medically validated information collected from patients during healthcare visits [4, 37, 38]. This study uses datasets for two diseases: coronavirus disease 2019 (COVID-19) for the clinical data and Ebola virus disease (EVD) for the tweet data.

2.1.1. Ebola virus disease dataset

The first dataset pertains to EVD, a highly infectious and often fatal illness caused by the Ebola virus. The dataset was released with the study of Mirugwe *et al.* [39], which reports 8,395 English-language tweets collected around the 2022 Ugandan Ebola outbreak using the Twitter Search application programming interface (API) through the Python Tweepy library. Keyword-based filtering was applied using search terms such as "Ebola", "Ebola virus", "Ebola outbreak", "EVD", "Ebola Uganda", and "Ebola crisis". The collected tweets encompass a broad range of content, including individual expressions of concern, official updates, healthcare-related commentary, and public awareness campaigns [10, 39].

The version of the tweet dataset analysed here contains 8,395 records after exact-duplicate removal on the timestamp, user and text fields. Its timestamps run from 3 January 2023 to 19 January 2023, a seventeen-day window that spans the closing phase of the Ugandan outbreak, which was declared over on 11 January 2023. Earlier drafts of this manuscript quoted a dataset of 13,629 tweets collected between 20 September and 30 November 2022; both figures were incorrect. The 13,629 count is the pre-deduplication size, and the 2022 window describes the outbreak rather than the collection period of the file analysed here. No further records were discarded: tweets whose cleaned text was empty after normalization were retained as empty strings rather than dropped, so the count entering feature extraction equals the count in Table 1.

2.1.2. Coronavirus disease (COVID-19) dataset

The COVID-19 clinical dataset used in this study was obtained from the study by Melchane *et al.* [40]. The data were sourced from the Albert Einstein Hospital in São Paulo, Brazil. The distributed file holds 5,644 records across 111 columns: a patient identifier, the target, and 109 candidate predictors. It comprises anonymized clinical records of patients who received care at the hospital. During their visits, biological samples were collected from these individuals for severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) reverse-transcription polymerase chain reaction (RT-PCR) testing, along with a range of additional laboratory diagnostics. To ensure patient confidentiality, all data were anonymized in accordance with internationally recognized standards and ethical guidelines.

Table 1: Composition of the Ebola tweet dataset after preprocessing. Class labels are assigned by the symptom dictionary of Section 2.2.1, not by human annotation. Partition sizes are not listed because all results come from repeated cross-validation rather than a single split.

Property	Value
Records after duplicate removal	8,395
Language	English
Collection window	3–19 Jan 2023
Non-warning records (class 0)	7,106
Warning records (class 1)	1,289
Positive prevalence	15.4%
Empty records after cleaning	64
Maximum length after cleaning	18 tokens
Symptom-dictionary terms	22
Tokenizer vocabulary retained	5,000

The variable groups are summarised in Table 2. Laboratory values were released already standardised to zero mean and unit variance by the data provider, and a large share of them are missing, since each patient received only the subset of assays their clinician ordered. That sparsity is a defining property of the dataset rather than a data-quality defect, and it turns out to govern what performance is attainable, so two cohorts are analysed throughout and reported separately. The *full cohort* comprises all 5,644 records with 102 predictors and median imputation of missing laboratory values; 558 records (9.9%) are positive. The *complete-case cohort* retains the 18 laboratory and care-pathway variables recorded for at least 10% of patients and then keeps only the 598 records complete on all of them, of which 81 (13.5%) are positive. The complete-case cohort is the one on which a usable model can be fitted, but restricting to it selects patients who received a full blood panel, and Section 3 quantifies how much of the apparent performance that selection supplies.

2.2. Dataset preprocessing

Dataset preprocessing is a critical step in machine learning, particularly when working with unstructured data such as tweets, as it transforms raw, noisy, and inconsistent data into a clean, structured format that models can effectively learn from [41, 42]. Similarly, structured data can be further preprocessed by eliminating redundant fields and addressing records with missing values. Without preprocessing, models may struggle with irrelevant noise, inconsistent data, or uninformative features, leading to poor performance, overfitting, or biased outcomes [42, 43]. The following subsections outline the preprocessing steps used to prepare the datasets for subsequent analysis.

Table 2: Variable groups in the COVID-19 clinical dataset (5,644 records, 102 predictors after removal of uniformly empty columns, and one label).

Group	Examples	Type
Demographic	Patient age quantile	Ordinal, 0–19
Care pathway	Regular ward, semi-intensive unit, intensive care unit admission	Binary
Red cell indices	Hematocrit, hemoglobin, mean corpuscular volume (MCV), mean corpuscular hemoglobin (MCH), mean corpuscular hemoglobin concentration (MCHC), red cell distribution width (RDW)	Continuous, standardised
White cell and platelet indices	Leukocytes, lymphocytes, monocytes, eosinophils, basophils, platelets	Continuous, standardised
Inflammatory marker	C-reactive protein	Continuous, standardised
Respiratory pathogen panel	Influenza A and B, Rhinovirus/Enterovirus, Coronavirus 229E and NL63, Parainfluenza 1–4, *Chlamydophila pneumoniae*	Categorical, detected / not detected / not tested
Target label	SARS-CoV-2 RT-PCR exam result	Binary

2.2.1. *Preprocessing of the unstructured dataset*

This section presents the workflow for preprocessing the unstructured tweet dataset. As depicted in Figure 1, the process begins with the extraction of disease-related tweets from the literature. The raw dataset, however, is often noisy and unstructured, containing irrelevant information, inconsistencies, and formatting issues, necessitating a series of preprocessing steps to make it suitable for ML model training.

The first preprocessing step is data cleaning, which focuses on removing noise and standardizing the text. This involves eliminating redundant elements such as URLs, emojis, and special characters such as @ and #, which are common in tweets but do not contribute to meaningful analysis. Additionally, stop words which are frequently occurring words such as "the," "is," or "and" that carry little semantic value are removed, along with punctuation, usernames, and numeric values. This cleaning process ensures that the dataset is streamlined, containing only the textual content that is relevant for downstream analysis. Equation (1) shows a representation of the raw tweet dataset, (2) presents the data cleaning function, and (3) shows the cleaned dataset.

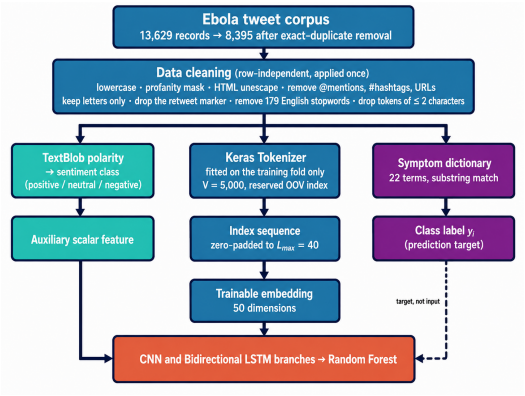

Figure 1: Workflow for preprocessing the tweet dataset. The symptom dictionary produces the prediction target; it does not contribute to the input vector.

$$T = \{t_1, t_2, \ldots, t_N\}, \tag{1}$$

$$t_i^{(c)} = C(t_i), \tag{2}$$

$$T^{(c)} = \{t_1^{(c)}, t_2^{(c)}, \ldots, t_N^{(c)}\}, \tag{3}$$

where t_i is a disease-related tweet extracted from literature or public sources and $C(\cdot)$ removes noise such as URLs, stopwords, and usernames.

Following data cleaning, the cleaned text is tokenized (Equation (4)), which breaks it into individual word tokens so that it can be analyzed systematically. Equation (5) defines a lemmatization step $l(\cdot)$ that would reduce each token to its dictionary form, for instance converting "running" to "run". It should be stated plainly that this step is *disabled* in the released implementation: the lemmatizer is constructed but the call is commented out, so $l(\cdot)$ is the identity in every result reported here, and the vocabulary consequently retains inflected forms as distinct terms. Equation (5) is retained because it describes the intended pipeline and because re-enabling the step is a one-line change, but no claim in this paper depends on it. No stemming is applied at any point.

$$t_i^{(tok)} = \{w_{i1}, w_{i2}, \ldots, w_{iL_i}\}, \tag{4}$$

$$t_i^{(lem)} = \{l(w_{i1}), l(w_{i2}), \ldots, l(w_{iL_i})\}, \tag{5}$$

where L_i is the number of tokens in tweet i and $l(\cdot)$ is the lemmatization function.

Parallel to these text-normalisation steps, which are required for sentiment analysis, the workflow incorporates preprocessing steps for symptom analysis and tweet labelling using a disease-symptom dictionary. Sentiment analysis (Equations (6) and (7)) categorizes tweets based on their emotional tone using the lexicon-and-rule-based polarity score of the TextBlob library, in which tweets with polarity greater than 0 are classified as positive, tweets with polarity of 0 are neutral, and those less than 0 are negative. These can provide insights into public perceptions of diseases.

Symptom analysis (Equations (8) and (9)) identifies mentions of specific disease-related symptoms within the tweets, often using a predefined dictionary of symptoms to map terms to known medical conditions. This dictionary also aids in tweet labelling, where tweets are tagged with relevant disease or symptom categories, creating structured labels for supervised learning tasks. The symptom dictionary contains the following 22 terms, matched as case-insensitive substrings of the cleaned tweet: *sore, fever, fatigue, malaise, muscle pain, headache, throat, vomiting, diarrhoea, diarrhea, abdominal pain, rash, kidney, liver, ill, weak, platelets, rest, sick, fighting, numb, flu*.

It should be stated plainly that the class label of Equation (9) is a deterministic function of this dictionary and of the tweet text: a tweet is labelled a warning signal exactly when at least one dictionary term occurs in it. The labels are therefore programmatic rather than human-annotated, and the implications of that design choice for the interpretation of the reported scores are discussed in Section 3.4.

$$s_i = \text{Sentiment}(t_i^{(lem)}), \tag{6}$$

$$c_i = \begin{cases} \text{Positive}, & s_i > 0 \\ \text{Neutral}, & s_i = 0 \\ \text{Negative}, & s_i < 0 \end{cases}, \tag{7}$$

$$z_i^{(sym)} = [\, I(d_1 \in t_i), \ldots, I(d_k \in t_i) \,], \tag{8}$$

$$y_i = \begin{cases} 1, & \sum_{k=1}^{K} I(d_k \in t_i) \geq 1 \\ 0, & \text{otherwise} \end{cases}, \tag{9}$$

where $d_1, d_2, \ldots, d_k$ are the disease symptoms, $z_i^{(sym)}$ is the symptom-indicator vector for tweet i, $I(\cdot)$ is the indicator function, and y_i denotes the tweet label.

The workflow then ranks the dataset vocabulary by term frequency using Equation (10), which both identifies the key terms dominating the discourse around the disease and determines which terms are retained. Feature generation does *not* use a bag-of-words representation. Each cleaned tweet is converted into an ordered sequence of integer word indices: a vocabulary is fitted over the training partition and truncated to the V most frequent terms, every retained term w_j is assigned an index $\iota(w_j) \in \{2, \ldots, V+1\}$ in descending frequency order, and every term outside that vocabulary is mapped to a single reserved out-of-vocabulary index $\iota_{\text{oov}} = 1$ rather than being discarded. Word order is preserved throughout, which is what makes the sequence a valid input to a recurrent layer.

$$f(w_j) = \sum_{i=1}^{N} \sum_{k=1}^{L_i} I\Big(w_j = l(w_{ik})\Big), \tag{10}$$

$$x_i^{(seq)} = [\, \iota(l(w_{i1})),\ \iota(l(w_{i2})), \ldots, \iota(l(w_{iL_i})) \,], \tag{11}$$

where $f(w_j)$ is the total term frequency of w_j across the training partition, V is the retained vocabulary size, and $\iota(\cdot)$ maps a token to its vocabulary index or to ι_{oov}. The published implementation set $V = 100$. That value is retained here only as an ablation: the dataset contains far more than 100 distinct informative terms, several symptom-dictionary terms fall outside the hundred most frequent, and Section 3 shows that this single setting accounts for most of the performance previously reported. All results labelled *revised* use $V = 5000$.

The index sequences are then zero-padded at the end (Equation (12)) to a fixed length $L_{\max} = 40$, which exceeds the longest cleaned tweet in the dataset (18 tokens), so that every tweet is represented by an equal-length vector and no sequence is truncated; the padding index 0 is reserved and carries no term. Standardizing the vector lengths in this way is required because the convolutional and recurrent layers take fixed-shape inputs. The padded index sequence is passed to a trainable embedding layer of dimension 50, whose output is the actual input to the CNN and BiLSTM branches. The scalar sentiment class of Equation (7) is appended after feature extraction, not before (Equation (13) and Section 2.3).

$$x_i^{(pad)} = \begin{cases} x_i^{(seq)}, & L_i = L_{\max} \\ [\, x_i^{(seq)}, \underbrace{0, \ldots, 0}_{L_{\max} - L_i} \,], & L_i < L_{\max} \end{cases}, \tag{12}$$

$$x_i = [\, f_{\text{cnn}}(x_i^{(pad)}) \,\|\, f_{\text{lstm}}(x_i^{(pad)}) \,\|\, s_i \,], \tag{13}$$

where $f_{\text{cnn}}(\cdot)$ and $f_{\text{lstm}}(\cdot)$ denote the CNN and BiLSTM feature maps defined in Section 2.3 and s_i is the sentiment class. The class label y_i of Equation (9) is the prediction target and does not appear in the input vector; earlier drafts of Equation (13) included it in error, and no experiment in this work used the label as a predictor.

2.2.2. *Preprocessing of the structured dataset*

The first step in preprocessing any structured dataset is to extract the dataset from relevant sources such as publicly available databases and literature and inspect the data as illustrated in Figure 2. Once extracted, preliminary inspection is conducted to understand the structure of the dataset such as the types of data contained in each column. For example, the clinical dataset summarised in Table 2 contains variables such as "Patient age quantile", "SARS-CoV-2 exam result", "Hematocrit", and "Platelets", which appear to be numerical or categorical. Understanding the data types and the general layout helps in determining how each column should be processed.

Healthcare datasets like this one contain extensive missing data, here because each patient received only the assays their clinician ordered rather than the full panel. The procedure applied was as follows. First, any column whose entries were uniformly zero or missing was removed, as such a column carries no information; no other column was dropped, and no missingness threshold was applied. Second, categorical columns, which are the respiratory pathogen assays and the target, were filled with a constant and label-encoded. Third, the remaining numerical columns were filled with the column median. Fourth, the target "SARS-CoV-2 exam result" was mapped to 1 for positive and 0 for negative. Seven columns are uniformly zero or missing and are removed by the first step, leaving 102 predictors of which 35 are categorical.

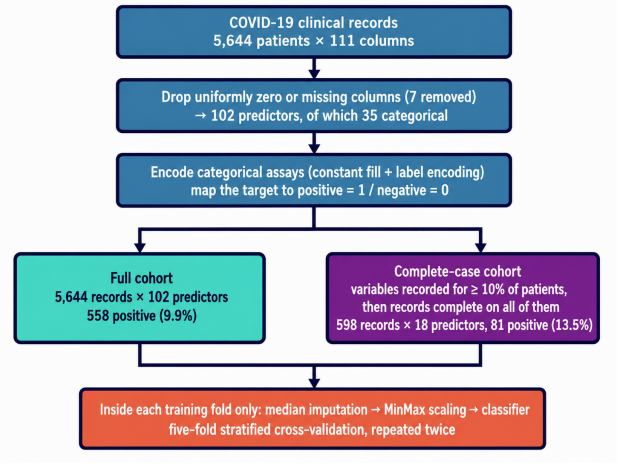

Figure 2: Workflow for preprocessing the clinical records, and the definition of the two cohorts. Imputation and scaling are fitted inside each training fold.

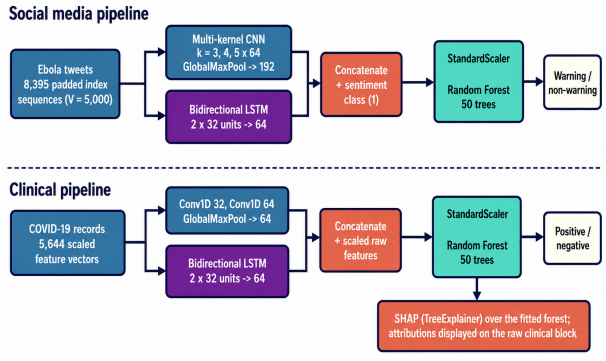

Figure 3: The proposed ensemble. Both modalities pass through a convolutional and a bidirectional recurrent branch in parallel; the concatenated feature maps, with modality-specific auxiliary features, are classified by a Random Forest. Both branches are trained end-to-end with a sigmoid head and then used as fixed feature maps for the forest (Section 2.4). The two pipelines share this template but use disjoint data on different diseases; they are trained and evaluated independently and are never fused into a single prediction.

One property of this procedure must be stated explicitly because it bears on how the results should be read: filling categorical assays with a constant makes "not detected" and "never ordered" indistinguishable to the model, so an apparent pathogen effect may in part encode which tests a clinician chose to request. This is also why the complete-case cohort is reported alongside the full one. In the published implementation the imputation and scaling statistics were estimated over the entire dataset before partitioning; in the present work both are refitted inside each training partition, and the results in Section 3 come from the corrected procedure. Class imbalance is handled by class-weighted forests rather than by resampling; the Synthetic Minority Over-sampling Technique (SMOTE) method cited in Ref. [41] is referenced as background and was not used.

Next, feature scaling is applied, which is essential when numerical values across columns have different ranges, as is common in clinical datasets. For instance, "Hematocrit" and "Mean platelet volume" might have vastly different scales, which could bias distance-based models. To avoid this, min–max normalisation, which scales values between 0 and 1, is applied as presented in Equation (14).

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}, \tag{14}$$

where x_{ij} is the original value of feature j for sample i and x'_{ij} is the normalized feature.

2.3. The ensemble framework

Figure 3 illustrates the proposed ensemble learning framework, which couples CNN and Bidirectional LSTM feature extraction with a Random Forest classifier. The framework is applied separately to two input types: social media text and structured clinical records. On the social media side, tweets are preprocessed as described in Section 2.2.1 and represented as padded sequences of vocabulary indices.

The same two-branch template is applied to both modalities, and this is the point on which earlier drafts of this manuscript were internally inconsistent. In each pipeline the input passes through *both* a convolutional branch and a bidirectional recurrent branch in parallel, and the two resulting feature maps are concatenated. On the social media side, the padded index sequence enters a shared embedding layer, from which a multi-kernel CNN extracts local n-gram patterns and a Bidirectional LSTM reads the same embedded sequence in both directions. On the clinical side, the scaled feature vector is reshaped along a single axis and passed to a convolutional branch and a Bidirectional LSTM branch in the same way. The clinical records are cross-sectional, so this axis is an ordering of features and not of time; the recurrent branch therefore models dependencies between adjacent feature positions rather than temporal dynamics, and no claim of temporal modelling is made for the clinical task.

Once both branches have produced their feature maps, the concatenated vector, together with the modality-specific auxiliary features, is standardized and used as the input to a Random Forest classifier. The Random Forest, composed of an ensemble of decision trees, is trained on these representations, enabling it to capture complex relationships by aggregating multiple tree-based predictions. During inference the trained Random Forest consumes the same representation to produce a final classification. The two pipelines are trained and evaluated entirely independently of one another, on datasets describing different diseases; at no point are social media and clinical evidence combined to produce a single prediction, and the framework should not be read as performing cross-modal fusion.

The feature composition operates as follows. For social media data, let f_{cnn} denote the 192-dimensional output of the multi-kernel CNN (three parallel filters of sizes 3, 4, and 5, each with 64 channels, followed by GlobalMaxPooling), and f_{lstm} the 64-dimensional output of the Bidirectional LSTM (2×32 hidden units). The scalar sentiment class s is appended, yield-

ing $x_{\text{combined}} = [f_{\text{cnn}} \| f_{\text{lstm}} \| s]$ with 257 dimensions. For clinical data, f_{cnn} is 64-dimensional (sequential Conv1D layers with 32 and 64 filters, kernel size 3), f_{lstm} is 64-dimensional (Bidirectional LSTM with 32 units), and every scaled clinical predictor is appended, so the combined width is 146 on the complete-case cohort (128 learned positions plus 18 predictors) and 230 on the full cohort (128 plus 102). No feature-selection procedure is applied at any point; an earlier draft quoted 16 raw clinical features and a combined width of 144, and neither figure describes the implementation. Both combined vectors are standardized with a StandardScaler fitted inside the training partition before being passed to the Random Forest classifier.

2.4. Training and evaluation protocol

The convolutional and recurrent branches are trained, which is a correction to the published implementation. In that implementation both branches were constructed and then called for prediction without ever being compiled or fitted, so their outputs were fixed random projections of the input at the initialization values, and only the Random Forest was trained. No optimizer, loss, or epoch count existed to report. In the present work each pipeline is assembled into a classifier by attaching a dropout layer ($p = 0.3$), a 32-unit ReLU dense layer, and a single sigmoid output, and is trained end-to-end with the Adam optimizer (learning rate 10^{-3}) under binary cross-entropy, with class weights inversely proportional to class frequency. Training uses a batch size of 64 on the social media task and 32 on the clinical task, a maximum of 15 and 40 epochs respectively, and early stopping on a 15% validation slice of the training partition with a patience of 3 and 6 epochs and restoration of the best weights. The trained branches are then detached at the concatenation layer and used as fixed feature maps for the Random Forest, which is the same two-stage arrangement the published design intends. Both the trained and the untrained variants are reported throughout, so the contribution of training is measurable rather than assumed.

Evaluation is by five-fold stratified cross-validation repeated twice, giving ten folds. Every fitted object, the tokenizer, the imputer, the scalers and the classifiers alike, is fitted inside the training partition of each fold and applied unchanged to the held-out partition; no statistic is estimated over the full dataset. The social media task is reported at the default threshold of 0.5 throughout. On the clinical task, the two fixed-specificity operating points use thresholds taken from the corresponding quantile of the out-of-bag predicted probabilities of the negative cases in each training partition; held-out data is never used to select a threshold. The Random Forest uses 50 trees on the social media task and 500 class-balanced trees on the clinical task, with a Gini splitting criterion, unlimited depth, and random_state = 42; Table 5 additionally reports the 50-tree clinical forest of the published implementation, which does not differ significantly from the 500-tree one. Reported scores are the mean over the ten folds, and the accompanying ± values are the standard deviation across folds, which is a real dispersion estimate rather than the unattributed figures that appeared in earlier drafts. Comparisons between configurations use the Wilcoxon signed-rank test on the paired per-fold scores.

This differs from approaches that use a single model structure, or that cast epidemic surveillance as a regression problem over case counts. The present framework is classification-oriented and is tailored to the identification of warning signals in individual records. Its distinguishing features are the pairing of convolutional and bidirectional recurrent feature extraction with a tree-ensemble decision stage, the application of one template to two structurally different modalities under a common evaluation protocol, and the use of explainable artificial intelligence to interpret the resulting predictions.

2.5. Computational complexity analysis

The computational complexity of the proposed framework is analysed for both training and inference phases. For the CNN component, each convolutional layer has training complexity $O(n{\cdot}L{\cdot}k{\cdot}d)$, where n is the number of samples, L is the sequence length, k is the kernel size, and d is the number of filters. GlobalMaxPooling adds $O(n{\cdot}L{\cdot}d)$. The Bidirectional LSTM has complexity $O(n{\cdot}L{\cdot}d_h^2)$ per direction, where d_h is the number of hidden units. The Random Forest training complexity is $O(n{\cdot}m{\cdot}T{\cdot}\log(n))$, where m is the feature dimension and T is the number of trees. For inference, CNN requires $O(L{\cdot}k{\cdot}d)$ per sample, LSTM requires $O(L{\cdot}d_h^2)$ per sample, and RF requires $O(T{\cdot}\text{depth})$ per sample. In practice, the total inference time per sample is dominated by the LSTM component but remains within milliseconds, making the framework suitable for real-time epidemic surveillance applications. By comparison, Transformer self-attention mechanisms have $O(L^2{\cdot}d)$ complexity per layer, which is quadratically more expensive with respect to sequence length.

3. Results and discussion

This section reports the experimental results for both tasks under the protocol of Section 2.4. The metrics reported are accuracy, positive-class sensitivity, specificity, positive-class precision and F1-score, the area under the receiver operating characteristic curve (AUROC), and, for the clinical task, average precision and sensitivity at fixed specificity. Macro-averaged recall is given alongside for comparability with earlier drafts, but is not used to support any claim.

The averaging convention matters here and is stated for every number. A macro-averaged recall is the unweighted mean of the recall of each class. On a problem where the majority class is classified almost perfectly it is dominated by that majority class and can remain high while the minority class is largely missed. Where an earlier version of this work reported "recall" without qualification, the value quoted was macro-averaged; the positive-class value, which is the quantity of interest in a screening setting, was in some cases an order of magnitude lower. Every score below is labelled, and the positive-class value is given first.

3.1. Predictive performance on social media data

Figure 4 shows the pooled confusion matrices for the published configuration and for the revised one, and Table 3 reports the full metric set for all seven configurations evaluated.

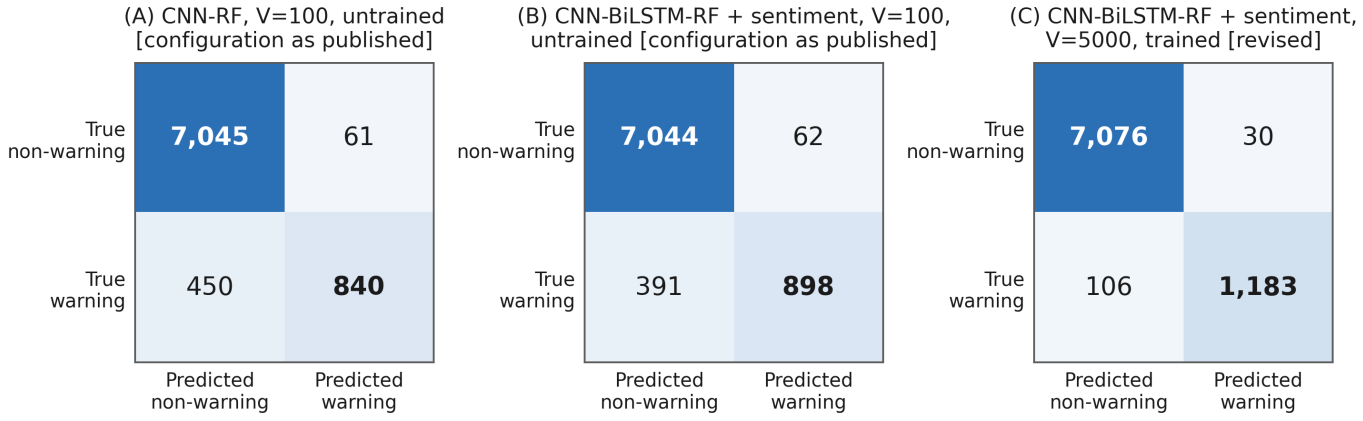

Figure 4: Social media task. Counts are summed over the ten cross-validation folds and divided by the number of repeats, so each panel totals the 8,395-record dataset. Panels A and B use the published configuration (100-term vocabulary, untrained branches); panel C is the revised configuration.

Table 3: Social media task: performance under five-fold stratified cross-validation repeated twice. Sensitivity, specificity, precision and F1-score refer to the positive (warning) class; ± values are the standard deviation across the ten folds. V is the tokenizer vocabulary size.

Configuration	Acc.	Sensitivity	Spec.	Precision	F1	AUROC	Macro rec.
CNN-RF, V=100, untrained *(as published)*	0.939	0.651±0.050	0.991	0.933±0.025	0.766±0.033	0.926±0.015	0.821
CNN-BiLSTM-RF + sent., V=100, untrained *(as published)*	0.946	0.697±0.037	0.991	0.935±0.020	0.798±0.026	0.938±0.012	0.844
CNN-BiLSTM-RF + sent., V=100, trained	0.956	0.767±0.033	0.990	0.934±0.019	0.842±0.020	0.940±0.010	0.879
CNN-RF, V=5000, trained	0.985	0.919±0.019	0.997	0.980±0.019	0.948±0.017	0.981±0.006	0.958
CNN-RF + sent., V=5000, trained	0.984	0.919±0.018	0.996	0.978±0.017	0.948±0.016	0.983±0.008	0.958
CNN-BiLSTM-RF + sent., V=5000, trained *(proposed)*	0.984	0.918±0.018	0.996	0.975±0.010	0.946±0.012	0.984±0.007	0.957
Term frequency–inverse document frequency (TF-IDF) + logistic regression *(reference)*	0.977	0.876±0.023	0.995	0.972±0.012	0.922±0.016	0.981±0.005	0.936

In the published configuration the framework recovers 651 of every thousand warning tweets at a precision of 0.933, for an F1-score of 0.766; adding the recurrent branch and the sentiment feature raises sensitivity to 0.697 and F1-score to 0.798. Both differences are significant across folds (p = 0.002, Wilcoxon signed-rank), and they are the improvements reported in earlier drafts. The revised configuration reaches a sensitivity of 0.918 and an F1-score of 0.946, an increase of 0.148 in F1-score over the published proposed model (p = 0.002). Accuracy moves only from 0.946 to 0.984 across the same comparison, which is why it is not treated as the headline result: 84.6% of the dataset belongs to the negative class, and a classifier that predicted "non-warning" everywhere would score 0.846 while detecting nothing.

3.1.1. *Where the improvement comes from*

Two changes separate the published configuration from the revised one, and an ablation over both shows that they, and not the ensemble structure, account for the whole of the gain.

Training the feature extractors is worth 0.044 in F1-score at the published vocabulary, raising the proposed model from 0.798 to 0.842 (p = 0.002). The tokenizer vocabulary is worth considerably more. Figure 5 sweeps it while holding everything else fixed: sensitivity rises from 0.767 at V = 100 to 0.875 at V = 1000 and 0.910 at V = 5000, and F1-score from 0.842 to 0.940. The mechanism is straightforward once stated. The

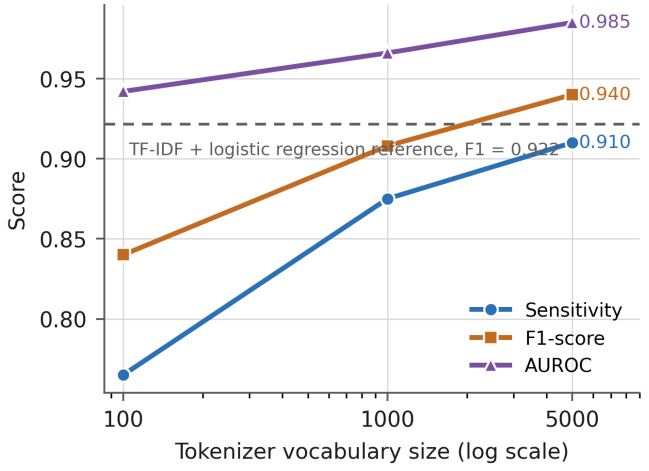

Figure 5: Effect of the tokenizer vocabulary cap on the proposed architecture, all else held constant. The dashed line is the term-frequency reference model of Table 3.

cleaned dataset contains several thousand distinct terms, and the symptom dictionary that defines the label has 22 entries, most of which do not fall among the hundred most frequent words in the dataset. A 100-term vocabulary therefore maps the majority of the label-bearing tokens to a single shared out-of-vocabulary index, and no amount of architectural elaboration downstream can recover information that the tokenizer has already discarded.

Once both corrections are made, the components that distinguish the proposed model from a plain convolutional pipeline stop contributing. At $V = 5000$ the CNN-RF model attains an F1-score of 0.948 and the full CNN-BiLSTM-RF model with sentiment attains 0.946; the difference is -0.003 in favour of the simpler model and is not significant ($p = 0.43$). Adding the sentiment feature alone likewise changes nothing ($p = 0.28$). The recurrent branch and the sentiment feature therefore earn their place only in the presence of the vocabulary bottleneck, where they partially compensate for information the tokenizer removed. This is a negative result about the architecture and it is reported as such: the ensemble is not harmful, but on this task, at an adequate vocabulary, it is not justified by its cost either.

3.1.2. *Comparison with a term-frequency reference*

Because the label is defined by a keyword rule, a linear model over term frequencies is the natural reference, and Table 3 includes one: TF-IDF features over unigrams and bigrams with class-balanced logistic regression. It attains a sensitivity of 0.876 and an F1-score of 0.922. The published configuration is well below this reference on every metric, by 0.124 in F1-score; the revised configuration is above it by 0.024 ($p = 0.002$). The comparison is worth stating plainly in both directions. It means the deep pipeline as originally reported was outperformed by a standard linear baseline that takes seconds to fit, and it also means that once the vocabulary and training defects are repaired the deep pipeline does exceed that baseline, though by a margin far smaller than the headline figures alone would suggest.

Table 4: Representative false negatives of the revised model on the social media task. Examples are paraphrased and redacted; handles are replaced by [user] and locations by [loc].

Pattern	Example
Metaphorical or political use	“[user] says he expects the fever to break tomorrow. Fever? This is political ebola.”
Incidental non-medical use	“Outbreak of verbal diarrhoea, it seems, and people are getting aggressive.”
Unhandled negation	“I don’t have Ebola but my colleague at work has been off sick for three days now with high fever.”
Short or sparse signal	“Cold sores again, maybe. Day 2 of this.”

3.1.3. *Error analysis of misclassified warning tweets*

On a single stratified held-out partition the revised model leaves 18 false negatives among 258 positive tweets. Every one of them contains exactly one symptom-dictionary term, and inspection shows that term is almost always used outside a medical context: metaphorical or political usage (“verbal diarrhoea”, “political ebola”), incidental usage, or a passing reference in an unrelated topic. Misclassified tweets are shorter than correctly classified positives, 8.9 tokens against 11.1 on average, giving a sparser representation. Negation patterns appear in 16.7% of them.

One pattern reported in earlier drafts does not survive the revised run. Neutral sentiment polarity accounts for 50.0% of the false negatives, but 50.3% of all positive tweets are neutral, so neutral sentiment is not over-represented among the errors and the earlier claim that “over 90%” of false negatives were neutral was an artefact of the weaker configuration. Table 4 gives illustrative examples, paraphrased and redacted. These findings still point toward context-aware embeddings and explicit negation handling, but the residual errors are now concentrated in genuinely ambiguous non-medical usage, which is a harder and more interesting problem than the one the original error analysis described.

3.2. *Predictive performance on clinical data*

The clinical task is reported on both cohorts defined in Section 2.1. Two reporting decisions govern how these numbers should be read, and both are corrections to earlier drafts.

The first concerns the decision threshold. A screening instrument is specified by its operating point, not by the arbitrary probability cut of 0.5, and on a task with 13.5% prevalence the default threshold sits far into the region where the classifier abstains from the minority class. Sensitivity is therefore reported at two clinically meaningful operating points, 90% and 80% specificity, alongside the default. The second concerns how those thresholds are obtained. A threshold read off the model’s own training-set scores does not transfer: a Random Forest classifies its training data almost perfectly, so the 90th

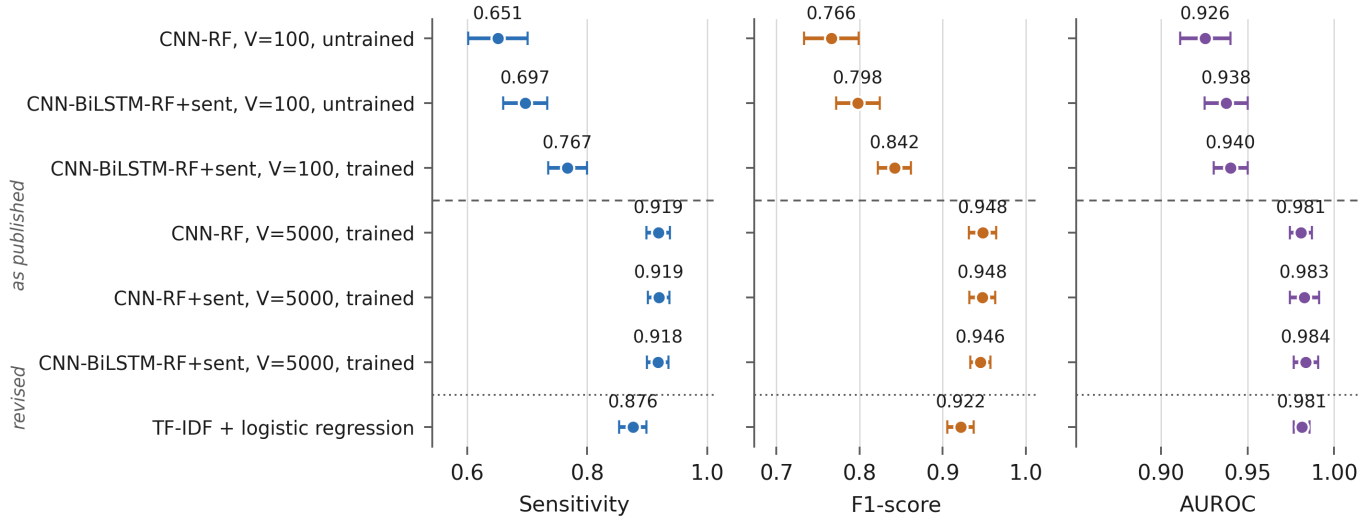

Figure 6: Social media task: mean and standard deviation across the ten folds. Configurations above the dashed rule use the published settings; those below use the revised settings, with the term-frequency reference at the foot.

percentile of its in-sample negative scores is far too low, and applying it to held-out data delivers 66% specificity rather than the intended 90%. All thresholds reported here are therefore estimated from out-of-bag predictions on the training partition only, which does transfer: the realised specificity on held-out data is 0.896 against a 0.90 target and 0.796 against a 0.80 target.

At the default threshold the recommended configuration recovers 31.0% of positive cases, which is the figure that would have been reported had the earlier convention been retained. At 90% specificity it recovers 71.1%, and at 80% specificity 80.3%. The model has not changed between these three rows of Table 5; only the operating point has. The discriminative ability described by the AUROC of 0.893 was present throughout and was concealed by reporting a single unsuitable threshold, and this is the single largest correction to the clinical results in this revision.

On the choice of classifier, the evidence is unambiguous and largely negative. A Random Forest on the laboratory variables alone is the strongest configuration on this cohort. Histogram gradient boosting is significantly worse (−0.030 in AUROC, $p = 0.027$; −0.124 in sensitivity at 90% specificity, $p = 0.004$), and the proposed CNN-BiLSTM ensemble is worse on both measures without reaching significance (−0.018 in AUROC, $p = 0.084$; −0.049 in sensitivity at 90% specificity, $p = 0.098$). Increasing the forest from the 50 trees of the published implementation to 500 changes nothing that survives testing (+0.006 in AUROC, $p = 0.57$). Nothing in the deep pipeline improves on a Random Forest fitted directly to eighteen laboratory variables, and Figure 8 shows the standard-deviation bands overlapping for every pair.

3.2.1. *The full cohort and what missingness costs*

On the full cohort of 5,644 records the picture is different in kind rather than degree. Median imputation followed by a Random Forest attains an AUROC of 0.617; replacing it with histogram gradient boosting, which routes missing values natively rather than filling them, attains 0.661, with average precision rising from 0.143 to 0.212 and sensitivity at 90% specificity from 0.115 to 0.180. Imputing a laboratory value that was never ordered destroys information that a model able to represent absence can use. Even so, the resulting discrimination remains weak.

3.2.2. *Cohort selection and a negative control*

The complete-case cohort is obtained by retaining variables recorded for at least 10% of patients and then keeping only records complete on all of them. A patient enters that cohort only if a clinician ordered a full blood panel, which is itself informed by clinical suspicion, so the obvious objection is that the model is recovering the ordering decision rather than the laboratory signal. That objection is testable, and Table 6 tests it: a classifier given *only* the number of assays ordered for each patient, and no laboratory value at all, attains an AUROC of 0.510, which is chance. The ordering decision carries essentially no information about SARS-CoV-2 status in this dataset, and the complete-case result is therefore not an artefact of who was tested.

What cohort restriction does supply is the laboratory variables themselves, and Figure 10 shows that this is an all-or-nothing property of the data. Relaxing the completeness threshold does not trade features against sample size smoothly. At thresholds of 6–10% the cohort holds 18 to 24 variables for 242 to 598 patients, and AUROC sits between 0.86 and 0.87. At 15% and above only four variables survive, but every one of the 5,644 records qualifies, and AUROC falls to 0.63. There is no intermediate cohort: this dataset consists of roughly six hundred patients who received a full blood panel and five thousand who received almost nothing.

Table 5: Clinical task, complete-case cohort (n = 598, 18 predictors, 81 positive). Five-fold stratified cross-validation repeated twice; ± values are the standard deviation across the ten folds. Operating-point thresholds are estimated from out-of-bag predictions on the training partition.

Configuration	AUROC	AP	Sens. @ 0.5	Sens. @ 90% spec.	Sens. @ 80% spec.
Random Forest on clinical features (recommended)	0.893±0.029	0.625	0.310±0.141	0.711±0.107	0.803±0.085
Random Forest, 50 trees *(as published)*	0.887±0.031	0.622	0.347±0.183	0.667±0.093	0.797±0.079
Histogram gradient boosting	0.863±0.037	0.599	0.636±0.116	0.587±0.108	0.710±0.097
CNN-BiLSTM-RF, trained *(proposed ensemble)*	0.875±0.029	0.665	0.383±0.154	0.661±0.109	0.802±0.058

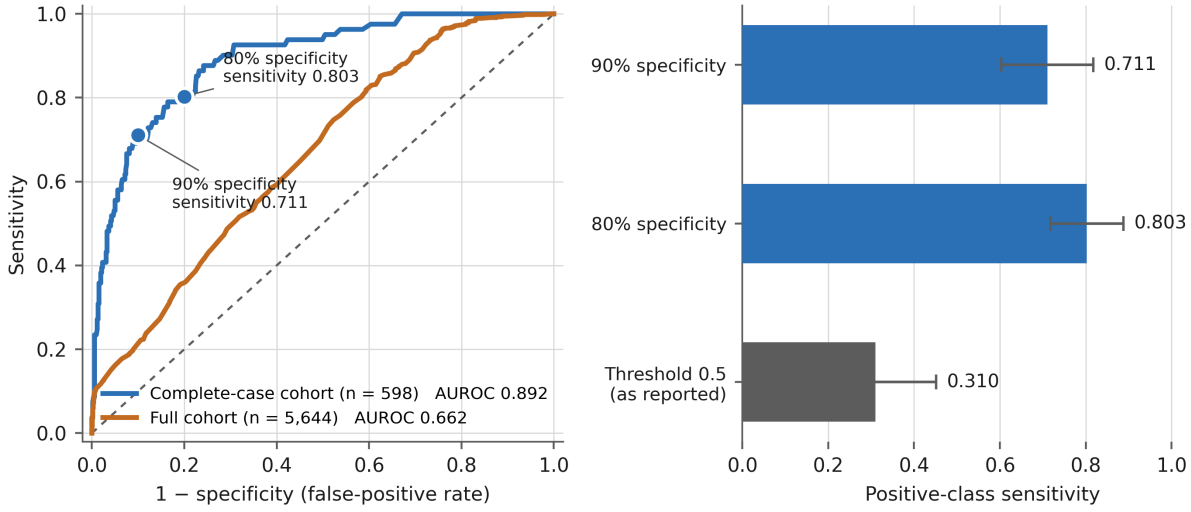

Figure 7: Clinical task. Left: receiver operating characteristic curves for the recommended configuration on each cohort, with the two operating points marked. Right: the same complete-case model reported at three thresholds.

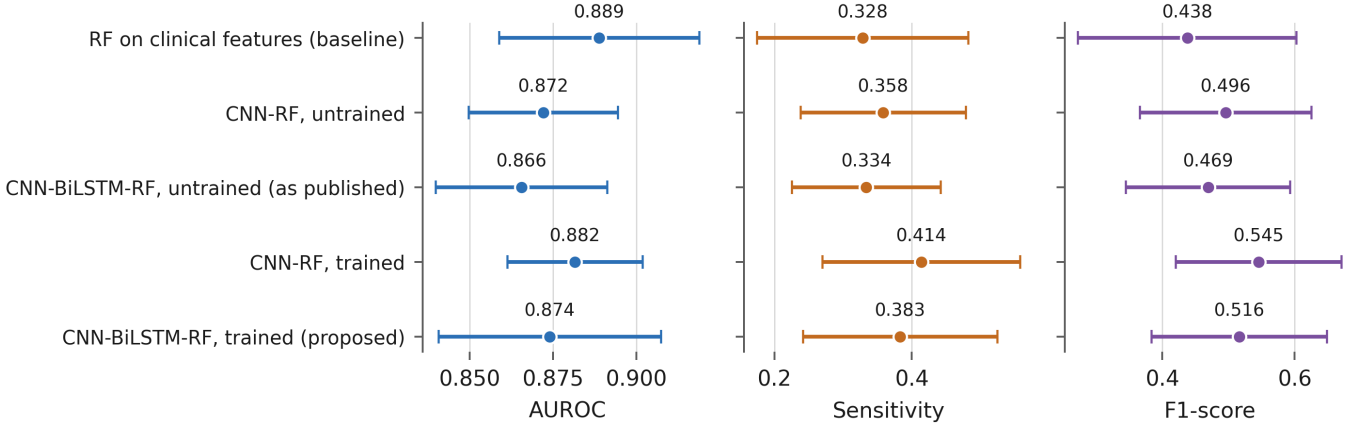

Figure 8: Clinical task, complete-case cohort: mean and standard deviation across the ten folds at the default threshold. The bands overlap for every pair.

3.2.3. What the model is, and is not, for

Taken together, these results define the instrument more precisely than earlier drafts did. A model that requires a complete blood count cannot be a point-of-care screening tool, because at the point of care the panel has not been ordered; on the population that has not been tested, discrimination is near

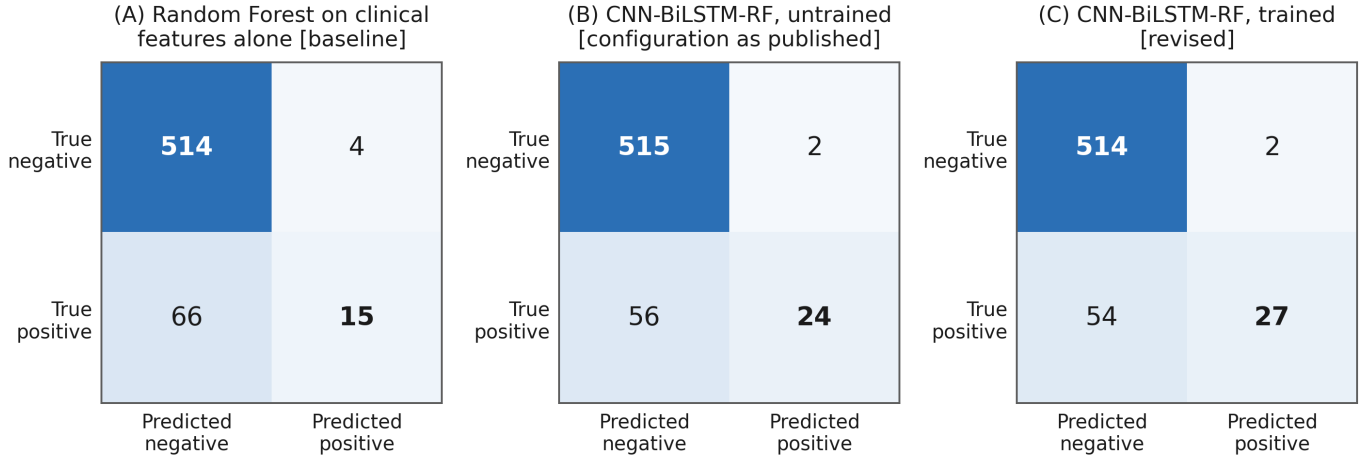

Figure 9: Clinical task, complete-case cohort (n = 598, 81 positive), decision threshold 0.5. Counts are pooled over the ten folds and divided by the number of repeats. Panel A is a Random Forest on the laboratory variables alone.

Table 6: Clinical task, full cohort (n = 5,644, 67 numeric predictors, 558 positive), same protocol. The negative control uses only the number of assays ordered for each patient and no laboratory value.

Configuration	AUROC	AP	Sens. @ 0.5	Sens. @ 90% spec.	Sens. @ 80% spec.
Random Forest + median imputation	0.617±0.018	0.143	0.581±0.063	0.115±0.033	0.246±0.034
Histogram gradient boosting, native missing-value handling (recommended)	0.661±0.012	0.212	0.799±0.060	0.180±0.102	0.277±0.080
Negative control: number of assays ordered, alone	0.510±0.022	0.111	0.068±0.027	0.073±0.026	0.073±0.026

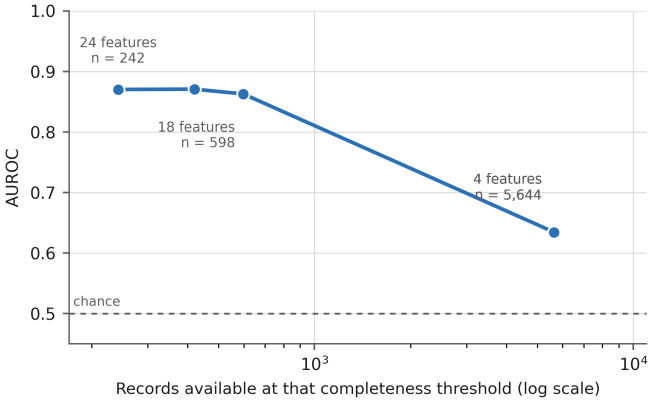

Figure 10: Discrimination against cohort size as the completeness threshold is relaxed. The feature availability is bimodal, and no intermediate cohort exists.

chance. But the complete-case cohort is not an arbitrary subset. It is the population for whom a blood count exists and a polymerase chain reaction result does not yet, and that gap is a real interval in the clinical pathway: a complete blood count returns within roughly half an hour, whereas reverse-transcription PCR takes hours to days.

Positioned there, as a post-haematology, pre-PCR triage aid rather than as a screening or diagnostic instrument, the framework is defensible on its own numbers. At 80% specificity it flags 80.3% of patients who will subsequently test positive, using a panel that has already been drawn, at no additional cost to the patient and with a result available while the confirmatory test is pending. It does not replace that confirmatory test, and the 19.7% of positive patients it misses at that operating point is the reason it cannot.

3.3. *Feature explainability with Shapley additive explanations (SHAP)*

The classifier's input is a concatenation of learned feature maps and raw clinical variables rather than clinical variables alone, so the mechanism by which the attributions are produced needs stating. TreeExplainer is applied to the fitted Random Forest over its full input vector, so the attributions belong to the deployed model and not to a surrogate. The plots then display only the trailing block of that vector, which is the scaled raw clinical features: the SHAP array is sliced along its feature axis to the final $|x_{\text{scaled}}|$ positions and along its output axis to the positive class, and paired with the matching columns of

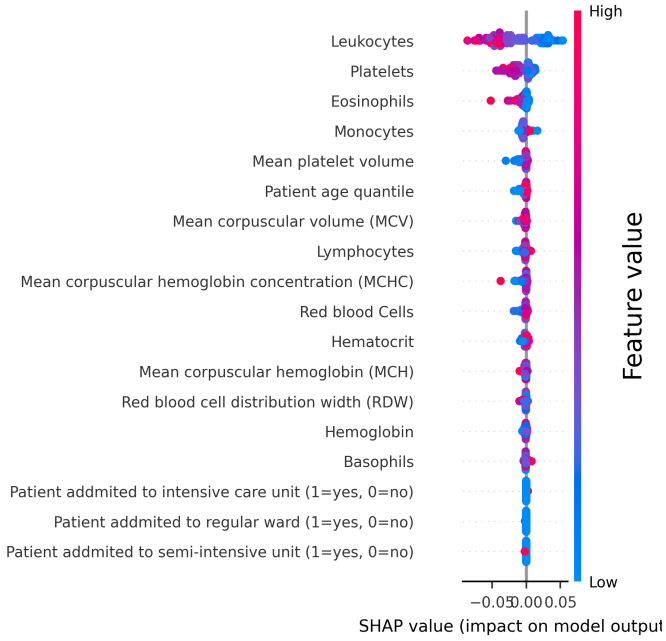

Figure 11: SHAP summary (beeswarm) plot over the raw clinical block of the classifier input, positive class, complete-case cohort.

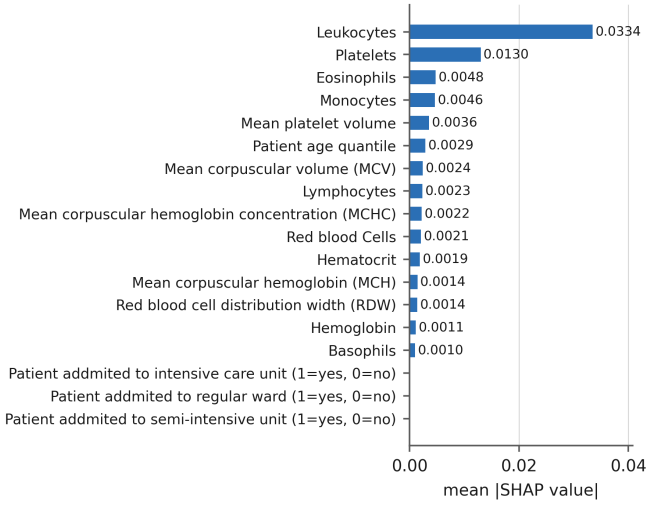

Figure 12: Mean absolute SHAP value per clinical variable. Together these account for 14.1% of the model's total attribution mass.

the held-out matrix and their variable names. Attributions carried by the convolutional and recurrent positions are not shown, because those positions have no clinical name to display.

That omission is larger than it may appear, and quantifying it is a necessary caveat on the interpretability claim. The raw clinical block accounts for 14.1% of the model's total mean absolute attribution; the remaining 85.9% sits in the learned positions, which are not interpretable in clinical terms. The plots below are therefore a faithful view of a minority of the model's behaviour, and the framework is interpretable in a weaker sense than an unqualified claim of explainability would imply.

Within that block the ranking is clear and is not the one reported in earlier drafts. Leukocytes dominate, with a mean absolute SHAP value of 0.0334, followed by platelets at 0.0130; eosinophils, monocytes, and mean platelet volume follow an order of magnitude lower. Patient age quantile ranks sixth at 0.0029, roughly a tenth of the leading variable. Earlier versions of this manuscript named patient age as the strongest driver of the model's predictions, and that claim does not survive the corrected analysis; it appears to have arisen from a run on a differently constructed feature set. Permutation importance computed on the held-out partition agrees on the direction: the largest single contributors are learned features, with platelets and eosinophils the highest-ranked named clinical variables.

The surviving interpretation is clinically coherent and more modest than before. The model responds principally to white-cell and platelet indices, which is consistent with the haematological signature of acute viral infection, and the care-pathway variables contribute almost nothing once the laboratory values are present. Because the attributions cover only a seventh of the model's behaviour, and because the model itself does not outperform a Random Forest on the same variables, these explanations should be read as a description of what the classifier attends to rather than as evidence for a diagnostic mechanism.

3.4. Limitations

The class labels on the social media task are programmatic rather than human-annotated. A tweet is labelled a warning signal exactly when one of 22 dictionary terms appears in it, and the classifier receives an encoding of the same text, so the task is the recovery of a deterministic keyword rule. The scores in Table 3 measure how closely the pipeline approximates that rule, not how well it identifies genuine epidemiological warning signals, and the high absolute values should be read in that light. A human-annotated evaluation subset is required before any claim of epidemiological validity can be made.

The architecture is not justified by the evidence presented. At an adequate vocabulary the recurrent branch and the sentiment feature produce no measurable improvement over a convolutional pipeline alone, and on the clinical task no configuration, deep or otherwise, differs significantly from a Random Forest on the laboratory variables. The framework's value on the social media task rests on training the extractors and on an adequate tokenizer vocabulary, both of which are ordinary implementation requirements rather than contributions.

The clinical instrument requires a complete blood count and is therefore not a point-of-care screening tool. On the population without one, discrimination is near chance (AUROC 0.66 at best), so the framework applies only in the interval between haematology and confirmatory testing. Even at its intended operating point it misses 19.7% of positive patients at 80% specificity, and 28.9% at 90% specificity, so it cannot substitute for the confirmatory test.

The complete-case cohort is a selected population of 598 patients from a single hospital, with only 81 positive cases.

Fold-to-fold variation in sensitivity is correspondingly wide (±0.107 at 90% specificity), and an operating point estimated on data of this size should be re-estimated on local data before any deployment.

The interpretability claim is partial. SHAP attributions over the named clinical variables account for 14.1% of the model's total attribution mass; the remainder lies in learned features with no clinical reading.

The application of a recurrent layer to the clinical dataset remains unconventional, as the records are cross-sectional. The Bidirectional LSTM models dependencies between adjacent positions in an arbitrary feature ordering, which is a pseudo-sequence and not a time series, and the ablations in Table 5 give no evidence that it helps.

Both datasets are narrow. The tweet dataset covers a seventeen-day window in January 2023 in English only, and the clinical records come from a single hospital in one city during one phase of one pandemic. Multi-site validation across diseases, languages and healthcare systems remains necessary.

4. Conclusion

This study evaluated an explainable CNN-BiLSTM-Random Forest framework on two independent epidemic surveillance classification tasks under repeated stratified cross-validation, with all preprocessing fitted inside the training partition and all metrics reported per class. Four conclusions follow from the evidence.

On the social media task the framework performs well in absolute terms, recovering 91.8% of warning tweets at a precision of 0.975 and an F1-score of 0.946, and exceeding a tuned term-frequency baseline by 0.024 in F1-score. This is a substantially better result than the same framework achieved as originally implemented.

The improvement, however, is attributable to two implementation corrections rather than to the architecture. Raising the tokenizer vocabulary from 100 to 5,000 terms and training the feature extractors end-to-end together account for the whole of the gain. Once both are in place, removing the recurrent branch and the sentiment feature costs nothing measurable, so the ensemble structure that the framework is named for is not supported by these experiments on this task.

On the clinical task the finding about the architecture is negative but the instrument itself is usable once specified correctly. No configuration improves on a Random Forest fitted to the eighteen laboratory variables alone, and the proposed ensemble is slightly worse. What changes the clinical conclusion is not the model but the operating point: at 90% specificity the recommended configuration recovers 71.1% of positive cases and at 80% specificity 80.3%, against 31.0% at the default threshold of 0.5. A negative control confirms that this is not an artefact of which patients were tested, since the number of assays ordered carries no predictive information on its own (AUROC 0.510). The instrument is best described as a post-haematology, pre-PCR triage aid rather than a screening or diagnostic tool, because on patients without a blood count discrimination is near chance.

On interpretability, SHAP attributions over the deployed classifier rank leukocytes and platelets highest, with patient age sixth, a pattern consistent with the haematological signature of acute viral infection. The attributions cover 14.1% of the model's behaviour, so the framework is interpretable in a partial sense that should be stated rather than assumed.

Taken together, the results support a narrower claim than the one this line of work began with. A convolutional feature extractor coupled to a tree ensemble, properly trained and given an adequate vocabulary, is an effective and interpretable classifier of textual warning signals; the additional recurrent and sentiment components, and the transfer of the whole template to sparse cross-sectional clinical records, are not supported by the evidence gathered here.

5. Future work

The most immediate extension is a human-annotated evaluation subset for the social media task, which would allow the pipeline to be assessed against epidemiological judgement rather than against a keyword rule, together with context-aware sentence embeddings and explicit negation handling to address the residual errors of Table 4. On the clinical side, the informative direction is not a larger architecture but better inputs: features available before a full blood panel is ordered, and a design that treats the ordering decision itself as data rather than as a filter. A genuine fusion experiment, on a dataset in which text and clinical evidence describe the same population over the same period, would test the multimodal claim that the present design cannot support, and prospective validation against dated outbreak endpoints would extend the work from classification toward early warning proper.

Deployment considerations are recorded here as prospects rather than results. The feature extraction pipeline has low inference latency, on the order of milliseconds per sample, and the modular architecture allows the branches to be parallelized on GPU hardware while the Random Forest scales linearly with data volume. Integration with surveillance platforms such as the Global Outbreak Alert and Response Network (GOARN) through RESTful application programming interfaces would be straightforward, and the compact model footprint suits edge deployment on low-power hardware through integer quantization and lightweight runtimes. Any deployment would need to comply with applicable health data regulations, and federated learning offers a route to training without centralizing patient records. None of these directions is evaluated in this work, and none should be pursued on the clinical task until the performance questions above are settled.

Data availability

The code, the fold-level cross-validation results, and the scripts that regenerate every figure and table in this paper are available at https://github.com/xacking/epidemic-warning-classification. The repository records the exact preprocessing, the 22-term symptom dictionary, the cross-validation seeds, the

definition of both clinical cohorts, and the out-of-bag procedure used to fix the reported operating points.

Neither dataset is redistributed. The Ebola tweet dataset was released with the study of Mirugwe *et al.* [39] and remains subject to the platform's terms, under which only tweet identifiers and a hydration route may be shared; the COVID-19 clinical records are publicly available in the Hospital Israelita Albert Einstein release used by Melchane *et al.* [40]. Instructions for obtaining both are given in the repository.

Declaration of competing interest

The authors declare no conflict of interest.

Funding

The authors state that no funding was involved.

Acknowledgment

The authors express sincere appreciation to the staff, faculty, and Head of the Department of Computer Science, Nile University of Nigeria, for their invaluable support and guidance throughout this study.

References

[1] H. Ren, Y. Ling, R. Cao, Z. Wang, Y. Li & T. Huang, "Early warning of emerging infectious diseases based on multimodal data", Biosafety and Health **5** (2023) 193. https://doi.org/10.1016/j.bsheal.2023.05.006.

[2] S. Moore, E. M. Hill, M. J. Tildesley, L. Dyson & M. J. Keeling, "Vaccination and non-pharmaceutical interventions for COVID-19: a mathematical modelling study", The Lancet Infectious Diseases **21** (2021) 793. https://doi.org/10.1016/S1473-3099(21)00143-2.

[3] J. Viana, C. H. van Dorp, A. Nunes, M. C. Gomes, M. van Boven, M. E. Kretzschmar, M. Veldhoen & G. Rozhnova, "Controlling the pandemic during the SARS-CoV-2 vaccination rollout", Nature Communications **12** (2021) 3674. https://doi.org/10.1038/s41467-021-23938-8.

[4] G. Giordano, M. Colaneri, A. Di Filippo, F. Blanchini, P. Bolzern, G. De Nicolao, P. Sacchi, P. Colaneri & R. Bruno, "Modeling vaccination rollouts, SARS-CoV-2 variants and the requirement for non-pharmaceutical interventions in Italy", Nature Medicine **27** (2021) 993. https://doi.org/10.1038/s41591-021-01334-5.

[5] G. Cho, J. R. Park, Y. Choi, H. Ahn & H. Lee, "Detection of COVID-19 epidemic outbreak using machine learning", Frontiers in Public Health **11** (2023) 1252357. https://doi.org/10.3389/fpubh.2023.1252357.

[6] A. AlArjani, M. T. Nasseef, S. M. Kamal, B. V. S. Rao, M. Mahmud & M. S. Uddin, "Application of mathematical modeling in prediction of COVID-19 transmission dynamics", Arabian Journal for Science and Engineering **47** (2022) 10163. https://doi.org/10.1007/s13369-021-06419-4.

[7] C. El Morr, D. Ozdemir, Y. Asdaah, A. Saab, Y. El-Lahib & E. S. Sokhn, "AI-based epidemic and pandemic early warning systems: a systematic scoping review", Health Informatics Journal **30** (2024) 14604582241275844. https://doi.org/10.1177/14604582241275844.

[8] R. Abdallah, S. A. AbdelGaber & H. A. Sayed, *Disease outbreak/epidemic in public health sector*, 6th International Conference on Computing and Informatics (ICCI 2024), Egypt, 2024, pp. 203–216. https://doi.org/10.1109/ICCI61671.2024.10485007.

[9] P. S. T. Shashank, A. Vaibhavi, J. Vaishnavi & M. A. Jabbar, "Regression model for prediction of epidemic outbreaks", IOP Conference Series: Materials Science and Engineering **1042** (2021) 012016. https://doi.org/10.1088/1757-899x/1042/1/012016.

[10] S. Amin, M. I. Uddin, D. H. alSaeed, A. Khan & M. Adnan, "Early detection of seasonal outbreaks from Twitter data using machine learning approaches", Complexity **2021** (2021) 5520366. https://doi.org/10.1155/2021/5520366.

[11] R. Aleixo, F. Kon, R. Rocha, M. S. Camargo & R. Y. De Camargo, *Predicting dengue outbreaks with explainable machine learning*, 22nd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid 2022), Taormina, Italy, 2022, pp. 940–947. https://doi.org/10.1109/CCGrid54584.2022.00114.

[12] D. Y. Kirange, V. M. Pathak & Y. N. Chaudhari, "Ensemble model for epidemic detection in Maharashtra using machine learning techniques", IOSR Journal of Computer Engineering **27** (2025) 31. https://www.iosrjournals.org/iosr-jce/papers/Vol27-issue2/Ser-1/F2702013138.pdf.

[13] B. Abdualgalil, S. Abraham & W. M. Ismael, "Early diagnosis for dengue disease prediction using efficient machine learning techniques based on clinical data", Journal of Robotics and Control (JRC) **3** (2022) 257. https://doi.org/10.18196/jrc.v3i3.14387.

[14] S. Sharma, Y. K. Gupta & A. K. Mishra, "Analysis and prediction of COVID-19 multivariate data using deep ensemble learning methods", International Journal of Environmental Research and Public Health **20** (2023) 5943. https://doi.org/10.3390/ijerph20115943.

[15] M. Abu Talib, Y. Afadar, Q. Nasir, A. Bou Nassif, H. Hijazi & A. Hasasneh, "A tree-based explainable AI model for early detection of Covid-19 using physiological data", BMC Medical Informatics and Decision Making **24** (2024) 179. https://doi.org/10.1186/s12911-024-02576-2.

[16] B. C. Bohm, F. E. de Melo Borges, S. C. M. Silva, A. T. Soares, D. D. Ferreira, V. S. Belo, J. S. Lignon & F. R. P. Bruhn, "Utilization of machine learning for dengue case screening", BMC Public Health **24** (2024) 1573. https://doi.org/10.1186/s12889-024-19083-8.

[17] S. Kumar, S. K. Gupta, V. Kumar, M. Kumar, M. K. Chaube & N. S. Naik, "Ensemble multimodal deep learning for early diagnosis and accurate classification of COVID-19", Computers and Electrical Engineering **103** (2022) 108396. https://doi.org/10.1016/j.compeleceng.2022.108396.

[18] F. O. Muhammed, A. O. Ogar, M. A. Suleiman & S. E. Abdullahi, "Machine learning for epidemic outbreak prediction: recent advances and open problems", NIPES Journal of Science and Technology Research **7** (2025) 642. https://doi.org/10.37933/nipes/7.4.2025.SI75.

[19] E. Y. Cramer *et al.*, "Evaluation of individual and ensemble probabilistic forecasts of COVID-19 mortality in the United States", Proceedings of the National Academy of Sciences **119** (2022) e2113561119. https://doi.org/10.1073/pnas.2113561119.

[20] A. Mahajan, N. Sharma, S. Aparicio-Obregon, H. Alyami, A. Alharbi, D. Anand, M. Sharma & N. Goyal, "A novel stacking-based deterministic ensemble model for infectious disease prediction", Mathematics **10** (2022) 1714. https://doi.org/10.3390/math10101714.

[21] P. K. Roy & A. Kumar, "Early prediction of COVID-19 using ensemble of transfer learning", Computers and Electrical Engineering **101** (2022) 108018. https://doi.org/10.1016/j.compeleceng.2022.108018.

[22] K. Sherratt *et al.*, "Predictive performance of multi-model ensemble forecasts of COVID-19 across European nations", eLife **12** (2023) e81916. https://doi.org/10.7554/eLife.81916.

[23] A. Sebastianelli, D. Spiller, R. Carmo, J. Wheeler, A. Nowakowski, L. V. Jacobson, D. Kim, H. Barlevi, Z. El Raiss Cordero, F. J. Colón-González, R. Lowe, S. L. Ullo & R. Schneider, "A reproducible ensemble machine learning approach to forecast dengue outbreaks", Scientific Reports **14** (2024) 3807. https://doi.org/10.1038/s41598-024-52796-9.

[24] Y. Liscano, L. A. Anillo Arrieta, J. F. Montenegro, D. Prieto-Alvarado & J. Ordóñez, "Early warning of infectious disease outbreaks using social media and digital data: a scoping review", International Journal of Environmental Research and Public Health **22** (2025) 1104. https://doi.org/10.3390/ijerph22071104.

[25] S. Wang, B. Z. Li, M. Khabsa, H. Fang & H. Ma, "Linformer: self-attention with linear complexity", arXiv preprint arXiv:2006.04768, 2020. https://arxiv.org/abs/2006.04768.

[26] I. Goodfellow, Y. Bengio & A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016. https://www.deeplearningbook.org.

[27] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal & S.-I. Lee, "From local explanations to global understanding with explainable AI for trees", Nature Machine Intelligence **2** (2020) 56. https://doi.org/10.1038/s42256-019-0138-9.

[28] N. Absar, N. Uddin, M. U. Khandaker & H. Ullah, "The efficacy of deep learning based LSTM model in forecasting the outbreak of contagious diseases", Infectious Disease Modelling **7** (2022) 170. https://doi.org/10.1016/j.idm.2021.12.005.

[29] X. Wen & W. Li, "Time series prediction based on LSTM-attention-LSTM model", IEEE Access **11** (2023) 48322. https://doi.org/10.1109/ACCESS.2023.3276628.

[30] S. F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. R. Varkonyi-Koczy, U. Reuter, T. Rabczuk & P. M. Atkinson, "COVID-19 outbreak prediction with machine learning", Algorithms **13** (2020) 249. https://doi.org/10.3390/a13100249.

[31] M. J. Kane, N. Price, M. Scotch & P. Rabinowitz, "Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks", BMC Bioinformatics **15** (2014) 276. https://doi.org/10.1186/1471-2105-15-276.

[32] H. Liany, A. Jeyasekharan & V. Rajan, "Predicting synthetic lethal interactions using heterogeneous data sources", Bioinformatics **36** (2020) 2209. https://doi.org/10.1093/bioinformatics/btz893.

[33] G. Wang, X. Liu, K. Wang, Y. Gao, G. Li, D. T. Baptista-Hon, X. H. Yang, K. Xue, W. H. Tai, Z. Jiang, L. Cheng, M. Fok, J. Y.-N. Lau, S. Yang, L. Lu, P. Zhang & K. Zhang, "Deep-learning-enabled protein-protein interaction analysis for prediction of SARS-CoV-2 infectivity and variant evolution", Nature Medicine **29** (2023) 2007. https://doi.org/10.1038/s41591-023-02483-5.

[34] A. I. Khan & A. Al-Badi, "Emerging data sources in decision making and AI", Procedia Computer Science **177** (2020) 318. https://doi.org/10.1016/j.procs.2020.10.042.

[35] S. Suma, R. Mehmood & A. Albeshri, "Automatic detection and validation of smart city events using HPC and Apache Spark platforms", in *Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies*, R. Mehmood, S. See, I. Katib & I. Chlamtac (Eds.), Springer, Cham, Switzerland, 2020, pp. 55–78. https://doi.org/10.1007/978-3-030-13705-2_3.

[36] Z. Shao, N. S. Sumari, A. Portnov, F. Ujoh, W. Musakwa & P. J. Mandela, "Urban sprawl and its impact on sustainable urban development: a combination of remote sensing and social media data", Geo-Spatial Information Science **24** (2021) 241. https://doi.org/10.1080/10095020.2020.1787800.

[37] X. Hu, C. Feng, T. Ling & M. Chen, "Deep learning frameworks for protein–protein interaction prediction", Computational and Structural Biotechnology Journal **20** (2022) 3223. https://doi.org/10.1016/j.csbj.2022.06.025.

[38] G. D. Maganga, J. Kapetshi, N. Berthet, B. K. Ilunga, F. Kabange, P. M. Kingebeni, V. Mondonge, J.-J. T. Muyembe, E. Bertherat, S. Briand, J. Cabore, A. Epelboin, P. Formenty, G. Kobinger, L. González-Angulo, I. Labouba, J.-C. Manuguerra, J.-M. Okwo-Bele, C. Dye & E. M. Leroy, "Ebola virus disease in the Democratic Republic of Congo", New England Journal of Medicine **371** (2014) 2083. https://doi.org/10.1056/NEJMoa1411099.

[39] A. Mirugwe, C. Ashaba, A. Namale, E. Akello, E. Bichetero, E. Kansiime & J. Nyirenda, "Sentiment analysis of social media data on Ebola outbreak using deep learning classifiers", Life **14** (2024) 708. https://doi.org/10.3390/life14060708.

[40] S. Melchane, Y. Elmir & F. Kacimi, "Infectious diseases prediction based on machine learning: the impact of data reduction using feature extraction techniques", Procedia Computer Science **239** (2024) 675. https://doi.org/10.1016/j.procs.2024.06.223.

[41] A. C. Flores, R. I. Icoy, C. F. Pena & K. D. Gorro, *An evaluation of SVM and naive Bayes with SMOTE on sentiment analysis data set*, 4th International Conference on Engineering, Applied Sciences and Technology (ICEAST 2018), Phuket, Thailand, 2018, pp. 1–4. https://doi.org/10.1109/ICEAST.2018.8434401.

[42] S. Arena, G. Manca, S. Murru, P. F. Orrù, R. Perna & D. Reforgiato Recupero, "Data science application for failure data management and failure prediction in the oil and gas industry: a case study", Applied Sciences **12** (2022) 10617. https://doi.org/10.3390/app122010617.

[43] K. Maharana, S. Mondal & B. Nemade, "A review: data pre-processing and data augmentation techniques", Global Transitions Proceedings **3** (2022) 91. https://doi.org/10.1016/j.gltp.2022.04.020.