

---

# AN ONLINE LEARNING APPROACH FOR DENGUE FEVER CLASSIFICATION

---

Siddharth Srivastava, Sumit Soman, Astha Rai

Centre for Development of Advanced Computing Noida, India  
{siddharthsrivastava, sumitsoman, asthar}@cdac.in

## ABSTRACT

This paper introduces a novel approach for dengue fever classification based on online learning paradigms. The proposed approach is suitable for practical implementation as it enables learning using only a few training samples. With time, the proposed approach is capable of learning incrementally from the data collected without need for retraining the model or redeployment of the prediction engine. Additionally, we also provide a comprehensive evaluation of machine learning methods for prediction of dengue fever. The input to the proposed pipeline comprises of recorded patient symptoms and diagnostic investigations. Offline classifier models have been employed to obtain baseline scores to establish that the feature set is optimal for classification of dengue. The primary benefit of the online detection model presented in the paper is that it has been established to effectively identify patients with high likelihood of dengue disease, and experiments on scalability in terms of number of training and test samples validate the use of the proposed model.

## 1 Introduction

The use of learning algorithms or models for disease detection and prediction has recently evoked substantial interest in the health informatics community as it can efficiently aid physicians in diagnosing diseases using data-driven models. These approaches are also useful for obtaining long-term occurrence trends and causative symptoms for various diseases by efficiently mining data from healthcare information systems. Such tasks would traditionally be intractable to execute by human intervention, however, the use of machine learning methods would provide credible results by efficient processing of available, annotated data.

The primary challenge in developing disease detection models is the availability of labelled data that has been validated by clinicians. It is essential that learning models for disease detection should be trained on adequate training samples in order to be able to generalize better for test samples. However, since obtaining a large amount of labelled training data is not always practically feasible, as it would take a long time to curate such a dataset, one can alternatively employ a learning model that is initially trained using a few labelled samples, but incrementally learns and updates the model as and when new labelled data is available. In other words, online learning models present a viable use-case in such situations. Our objective in this paper is to develop a viable online learning model for the task of dengue disease detection, which is a mosquito-borne tropical disease which is difficult to diagnose, and therefore leads to fatalities. We now present a brief survey of recent literature in this domain to familiarize the reader with the efforts taken by researchers in this direction, and motivate the context and novelty of our work.

Potts *et al.* [1] have used laboratory findings collected 72 hours after fever onset (including White Blood Cell (WBC) count, percent monocytes, platelet count and hematocrit) to classify severity of dengue disease among patients visiting two hospitals (one rural and one urban) in Thailand. Althouse *et al.* [2] used data retrieved from internet search for dengue occurrences in Singapore and Bangkok during 2004-2011 and evaluated multiple learning models to determine incidence of dengue. A similar effort using a larger dataset retrieved from internet search across the countries of Bolivia, Brazil, India, Indonesia and Singapore was presented in the work by Chan *et al.* [3]. Spatial and temporal models for dengue prediction have also been presented in the works by Dom *et al.* for Malaysia [4], Luz *et al.* for Brazil [5], Phung *et al.* for Vietnam [6], Rotela *et al.* for Argentina [7] among others.

On a global scale, Bhatt *et al.* [8] have provided analysis on a larger dataset using reported occurrences to develop a modeling framework for global dengue risk determination. Hales *et al.* [9] determine the effect of global climate change on vector borne diseases, specifically based on vapor pressure levels. In fact, climate based dengue disease prediction has appeared in other works as well such as that of Descloux *et al.* [10], Hii *et al.* [11], Pinto *et al.* [12], Earnest *et al.* [13].

Multiple learning approaches have also been employed for dengue detection, including Decision-Tree algorithms [14], neural networks [15], ensemble based methods [16], Adaptive Neuro-Fuzzy Inference System [17], among others. A significant number of approaches in literature have employed Support Vector Machines and their variants for dengue disease prediction. These include works by Wu *et al.* [18], Yusof *et al.* [19], Khan *et al.* [20], Gomes *et al.* [21], Radzol *et al.* [22] among others.

As can be seen, most efforts in literature have performed offline analysis on dengue datasets which have been reported, and it is evident that the detection of this disease from clinical datasets is of use to the clinicians as well as healthcare policy makers. In order to practically realize the benefits of a dengue detection system, it is important to have the model deployed in a healthcare information system where it can analyze data samples in real-time and flag samples with high likelihood of the disease to the users of the system. These can then aid the clinician to arrive at a diagnosis for specific patients. Moreover, it can be useful to data scientists and clinical researchers in identifying and evaluating the factors that contribute to efficient detection of dengue. In this paper, we obtain a novel dataset derived from a healthcare information system deployed across multiple healthcare facilities. The dataset is pre-processed to obtain a feature space which has a combination of patient symptoms and clinical investigations that are conventionally used to diagnose dengue. We first establish the adequacy of this feature space for dengue detection by evaluating a few classifier models and analyzing their scalability by varying the number of training and test samples. Subsequently, we proceed to develop an online dengue classification model which is trained in an online manner (the classifier models is updated as and when new samples with dengue as the diagnoses are validated by the physician). To this end, we evaluate multiple online classifier models and provide a comprehensive report of their performance. Our results demonstrate that the obtained feature representation and online classifier trained can be practically used in a healthcare information system for dengue detection.

The rest of the paper is organized as follows. Section 2 discusses the design and architecture of an online model for dengue classification that can be used in a healthcare information system. A discussion of the machine learning methods that have been analyzed on the dataset in this paper is presented in Section 3. Section 4 comprises of a description of the features used in our dataset along with results using the classifier models. Finally, conclusions and future work are presented in Section 5.

## 2 Methodology and System Design

We begin with a description of the design and architecture of the dengue classification system, shown in Fig. 1, that can be integrated into a healthcare information system for online predictions on streaming data samples. The pre-requisite for using such a model is that the system should record clinical data, including symptoms and results of laboratory investigations, for patients along with a suitable mechanism for recording diagnoses for a patient, which is mapped to patients via suitable identifiers [23, 24]. Once such a system is in place, a dataset generation and model training component can be incorporated into the healthcare information system. However, for this model to learn incrementally from new data samples, a diagnoses validation user interface also needs to be provided, which can validate the clinician's diagnoses with that arrived at by the learning model, in order to incrementally improve the learning model to deliver robust predictions.

A two-phase approach has been adopted in this work to develop such an online dengue prediction model. Initially, we extract the features from a small training set which comprises of annotated samples of patients diagnosed with dengue fever. The dataset is derived after multiple pre-processing steps which involve extracting the data samples from the relational database used by the healthcare information system, cleaning the dataset to extract the features, which may also include steps to estimate missing values as well as normalize the dataset. Using these small number of samples represented in an appropriate feature space, we train a classifier model to predict whether a patient is likely to be diagnosed with dengue, based on the symptoms presented that have been used to train the model.

It may be noted here that choosing the feature space in which the data samples are to be represented is also a critical task as the generalization ability of the model depends on the diversity of the chosen features. It is important to choose as many features as possible that could be significant in detecting dengue, while also keeping in mind that the features should be such that there should be high availability in the healthcare information system (which means that it should be pertinently recorded across samples), while also being accurate in terms of precision and representation formats. In an ideal scenario, symptoms, investigations and diagnoses should be recorded using healthcare standards such as

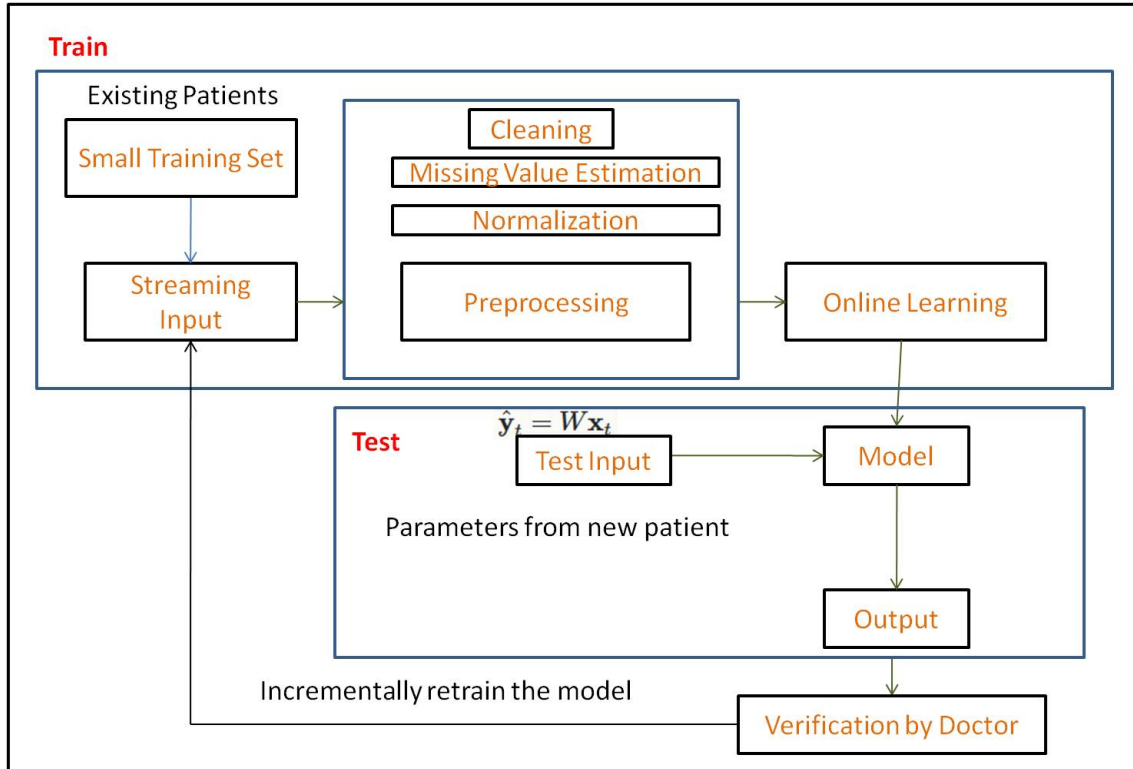


Figure 1: Dengue Detection System Architecture

SNOMED [25], LOINC [26], among others [27]. However, the integration of such standards into legacy hospital information systems is still in its infancy [28]. Therefore, in our work, we choose a set of features that are primarily categorical in nature and encompass the broad set of symptoms and laboratory investigations mandated for diagnosing dengue. This enables easier extraction and generation of data samples for classifier training, while also reducing the complexity of the dengue detection model. Moreover, the nature of features chosen makes the model ubiquitous for implementation across multiple healthcare information systems as well.

In the next phase of the dengue classification system, the trained classifier is used to obtain predictions on unseen samples (new patients whose symptoms and investigation results have been recorded and likelihood for dengue as a diagnosis needs to be evaluated). The predictions arrived at by the online learning model are then validated by clinicians. These labeled data samples are then used to re-train the classifier, in order to improve its generalization. Over a period of time, as more and more training data becomes available, the classifier training will become more robust and high accuracy predictions would be delivered by the dengue classification system.

In the following section, we discuss the learning models on which our dataset has been evaluated, which includes algorithms for offline and online classifier training.

### 3 Learning Algorithms Evaluated

In this section, we briefly describe the learning algorithms used to evaluate the generalization on the dataset. The idea is to be able to comprehensively evaluate how multiple learning online algorithms perform so that an optimal learning model could be used for online evaluation in applied systems.

A brief of the notation used in the following sub-sections is discussed for the convenience of the reader. We shall denote a data sample in  $n$  dimensions by  $x \in \mathbb{R}^n$ . The data matrix constitutes of  $M$  such samples and is denoted by  $X \in \mathbb{R}^{M \times n}$ . The labels for the samples are indicative of the class to which the sample belongs to, and for the case of binary classification (with two classes), is denoted as  $Y = \{y^i | y^i \in \{+1, -1\}, \forall i = 1, 2, \dots, M\}$ . The weight vector, which represents the decision boundary for identifying the class of a sample is denoted by  $w \in \mathbb{R}^n$ , and may also constitute of a bias term denoted by  $b \in \mathbb{R}$ . The weight vector is conventionally distributed normally as  $\mathbb{N}(\mu, \Sigma)$ , where  $\mu$  and  $\Sigma$  denote the mean and co-variance of the distribution respectively.

### 3.1 Offline Learning Algorithms

These algorithms have primarily been used to establish the adequacy of the feature set for the task of dengue disease classification. Baseline results have been obtained using the Support Vector Machine (SVM) and Random Forests, which have been discussed below.

#### 3.1.1 Support Vector Machine (SVM)

The Support Vector Machine (SVM) [29] for binary classification solves the following Quadratic Programming Problem (QPP).

$$\min_{w,b,\xi} \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^M \xi_i \quad (1)$$

subject to,

$$y^i (w^T x^i + b) + \xi_i \geq 1, \forall i = 1, 2, \dots, M \quad (2)$$

$$\xi_i \geq 0, \forall i = 1, 2, \dots, M. \quad (3)$$

Here, the separating hyperplane is  $w^T x + b = 0$  and  $\xi_i$  are the positively constrained slack variables.  $C$  is a hyperparameter that controls the tradeoff between minimizing the  $L_2$  norm of the weight vector and the slack variables. The class to which a test point  $x \in \mathbb{R}^n$  belongs is found by evaluating  $\text{sgn}(w^T x + b)$ . We mention here in passing that the SVM is largely popular owing to the kernel trick which allows mapping the input features to a high-dimensional space where a linear classifier can be found, and that the dual formulation of (1)-(3) is practically used owing to its tractability.

#### 3.1.2 Random Forests

Random forests, or random decision trees [30] consist of generating multiple decision trees and then using their combination to arrive at a prediction on a test sample. Each decision tree is trained on a subset of the training samples drawn with replacement from the training dataset. Random forests are also trained using a subset of features of the dataset instead of the entire dataset. Multiple variants of this method are also available in literature.

### 3.2 Online Learning Algorithms

We evaluate multiple online learning algorithms, which are briefly summarized in the following subsections. The general working mechanism of the online learning algorithms is as follows. Each algorithm suitably initializes the model parameters  $w$  and  $b$  and stochastically updates these parameters based on different update rules. These update rules are often dependent on evaluation of a suitable loss function, which is computed by the predicted label and the true label of the incoming sample. It is expected that the training will converge to the optimal classifier model parameters as it is trained stochastically.

#### 3.2.1 Adaptive Regularization Of Weights (AROW) and its variants

The Adaptive Regularization Of Weights (AROW) [31] uses adaptive regularization on the weight vector with each training sample in order to make the learning model robust to make it robust to label noise. The algorithm begins by initializing  $w = 0$  and  $\Sigma = I$ , where  $I$  denotes the identity matrix. For each sample, the weight vector is initially updated by the mean  $\mu^i$  and the prediction on the sample is computed as  $\text{sgn}(w^T x^i)$ . Based on the true label  $y^i$ , the loss is computed using (4), and the model parameters are updated as given by (5)-(6) when the computed loss term is positive.

$$l_i = \max\{0, (1 - y^i w^T x^i)\} \quad (4)$$

$$\mu_{i+1} = \mu_i + \alpha_i \sum_i y^i x^i \quad (5)$$

$$\Sigma_{i+1} = \Sigma_i + \beta_i \sum_i x^i x^{iT} \Sigma_i \quad (6)$$

where  $\alpha_i = l_i(\mu_i, (x^i, y^i))\beta_i$  and  $\beta_i = \frac{1}{x^{iT}\Sigma_i x^i + \gamma}$  for a suitable hyper-parameter  $\gamma$ .

A variant of the AROW, the New Adaptive Regularization of Weights (NAROW) uses another additional second order term which restricts the eigenvalues of  $\Sigma$  within specific bounds. The update rules for NAROW are given by (7)-(10), where  $b > 0$  is a suitably chosen bound.

$$\mu_{i+1} = \mu_i + y^i x^i \quad (7)$$

$$\Sigma_{i+1} = \Sigma_i + \beta_i \sum_i x^i x^{iT} \Sigma_i \quad (8)$$

$$\text{where, } \beta_i = \frac{1}{x^{iT} \Sigma_i x^i + \gamma_i} \quad (9)$$

$$\gamma_t = \begin{cases} \frac{x^{iT} \Sigma_i x^i}{bx^{iT} \Sigma_i x^{i-1}}, & x^{iT} \Sigma_i x^i \geq \frac{1}{b} \\ +\infty, & \text{otherwise} \end{cases} \quad (10)$$

### 3.2.2 Online Gradient Descent

The Online Gradient Descent (OGD) [32] is a stochastic or batch-wise update version of the classical gradient descent algorithm for optimization. The approach is similar to the AROW except for the weight update rule, which is given by (11) for OGD.

$$w_{i+1} = w_i + l_i \sqrt{\frac{1}{iy^i x^i}} \quad (11)$$

In principle, the OGD combines conventional gradient descent with the hinge loss function. The stochastic update rules makes it amenable for large datasets.

### 3.2.3 Confidence-weighted learning (CW) and Soft variants (SCW 1 & SCW 2)

Confidence-weighted learning aims at updating the weight distribution using the Kullback-Leibler divergence ( $D_{KL}$ ) with the constraint that the probability of correct classification is constrained by a determined threshold  $\eta$ . This involves the update rule given by the optimization problem in (12)-(13).

$$(\mu_{i+1}, \Sigma_{i+1}) = \arg \min_{\mu, \Sigma} D_{KL}(\mathbb{N}(\mu, \Sigma) \| \mathbb{N}(\mu_i, \Sigma_i)) \quad (12)$$

subject to

$$P_{w \in \mathbb{N}(\mu, \Sigma)}(y^i [w^T x^i] \geq 0) \geq \eta \quad (13)$$

The SCW improves upon the CW and AROW by the use of a modified loss function which is given by (14) for SCW 1 and (15) for SCW 2.

$$(\mu_{i+1}, \Sigma_{i+1}) = \arg \min_{\mu, \Sigma} D_{KL}(\mathbb{N}(\mu, \Sigma) \| \mathbb{N}(\mu_i, \Sigma_i)) + Cl^\phi(\mathbb{N}((\mu, \Sigma); (x^i, y^i))) \quad (14)$$

$$(\mu_{i+1}, \Sigma_{i+1}) = \arg \min_{\mu, \Sigma} D_{KL}(\mathbb{N}(\mu, \Sigma) \| \mathbb{N}(\mu_i, \Sigma_i)) + Cl^\phi(\mathbb{N}((\mu, \Sigma); (x^i, y^i)))^2 \quad (15)$$

where  $C$  is a hyper-parameter and  $l^\phi(\mathbb{N}((\mu, \Sigma); (x^i, y^i))) = \max(0, \phi \sqrt{x^{iT} \Sigma x^i} - y^i \mu x^i)$ . Here,  $C$  is a hyperparameter that determines the relative weight to be assigned to the terms in the update rule.

### 3.2.4 Normalized HERD (NHERD)

The Normalized HERD (NHERD) [33] uses the loss function defined by (16)-(18), in order to herd a normalized weight distribution by constraining the velocity flow. It may be noted here that  $\mathbb{E}$  denotes the expectation operator.

$$\mu_{i+1} = A_i \mu_i + b_i \quad (16)$$

$$\Sigma_{i+1} = A_i \Sigma_i A_i^T \quad (17)$$

$$\text{where, } (A_i, b_i) = \arg \min_{A, b} \mathbb{E}_{w \in \mathcal{N}(\mu_i, \Sigma_i)} C_i(A_i w^i + b_i) \quad (18)$$

$$C_i(w) = \frac{1}{2} [(w - w^i)^T \Sigma_i^{-1} (w - w^i) + C \max(0, 1 - y^i w^T x^i)]^2 \quad (19)$$

### 3.2.5 PA and its variants PA1 & PA2

Passive Aggressive (PA) [34] learning methods update the weight of the classifier model in the current iteration by minimizing the loss suffered by the classifier on the current sample. This effectively minimizes the error obtained by the classifier model trained in the previous iteration. The update rules are given by (20)-(21).

$$w_{i+1} = \arg \min_w \frac{1}{2} \|w - w^i\|^2 \quad (20)$$

$$\text{subject to, } \max(0, 1 - y^i w^T x^i) = 0 \quad (21)$$

Extensions to PA include PA1 and PA2, whose objective functions are given by (22) and (23) respectively.

$$w_{i+1} = \arg \min_w \frac{1}{2} \|w - w^i\|^2 + C \max(0, 1 - y^i w^T x^i) \quad (22)$$

$$w_{i+1} = \arg \min_w \frac{1}{2} \|w - w^i\|^2 + C \max(0, 1 - y^i w^T x^i)^2 \quad (23)$$

Here,  $C > 0$  is a suitably chosen hyperparameter to control the weight assigned to the individual terms in the update rule. It can be seen that PA1 is an unconstrained version of PA which can directly be solved by gradient descent based methods, and hence is suitable for larger datasets. PA2 is similar to PA1, except that it uses the squared loss function.

### 3.2.6 Improved Ellipsoid Method (IELLIP)

The Improved Ellipsoid Method (IELLIP) [35] uses the update rules as given by (24)-(27).

$$\mu_{i+1} = \mu_i + \alpha_i \sum_i g_i \quad (24)$$

$$\Sigma_{i+1} = \frac{1}{1 - c_i} (\Sigma_i c_i \sum_i g_i g_i^T \Sigma_i) \quad (25)$$

$$\text{where, } \alpha_i = \frac{\alpha \gamma - y^i \mu_i^T x^i}{\sqrt{x^{iT} \Sigma_i x^i}} \quad (26)$$

$$g_i = \frac{y^i x^i}{\sqrt{x^{iT} \Sigma_i x^i}}, c_i = cb^T, 0 \leq c, b \leq 1 \quad (27)$$

### 3.2.7 Approximate Large Margin Algorithm (ALMA)

The Approximate Large Margin Algorithm (ALMA) [36] implements an alternative loss function characterized by parameter  $\alpha$  in order to obtain a classifier model with margin dependent on the parameter. The update rule is given by (28)-(29).

$$w = \frac{w + l_i \sqrt{\frac{2}{k}} y^i \frac{x^i}{\|x^i\|}}{\max(1, \|w + l_i \sqrt{\frac{2}{k}} y^i \frac{x^i}{\|x^i\|}\|)} \quad (28)$$

$$\text{where, } l_i = \max(0, \frac{1 - \alpha}{\alpha \sqrt{k}} - \frac{y^i w^T x^i}{\|x^i\|}), k = k + l_i \quad (29)$$

### 3.2.8 Perceptron and Second Order Perceptron (SOP)

The classical perceptron [37] learning model is given by (30)-(31). It simply updates the classifier weights by a loss function dependent upon the number of misclassified samples.

$$w^{i+1} = w^i + l_i y^i x^i \quad (30)$$

$$\text{where, } l_i = \sum_i (y_{pred}^i \neq y^i) \quad (31)$$

Here,  $y_{pred}^i$  denotes the predicted label on the sample  $x^i$  and  $y^i$  is the true label of the sample. A variant of the perceptron, the Second Order Perceptron (SOP) [38], works in a similar manner while also additionally updating the covariance matrix  $\Sigma$  by a factor of  $\alpha I$ , where  $\alpha > 0$  is a suitably chosen hyperparameter and  $I$  is the identity matrix.

### 3.3 Relaxed Online Maximum Margin Algorithm (ROMMA) & aggressive ROMMA (aROMMA)

The Relaxed Online Maximum Margin Algorithm (ROMMA) [39] uses an alternate weight update rule as given by (32).

$$w_{i+1} = \frac{\|x^i\|^2 \|w^i\|^2 - y^i w^T x^i}{\|x^i\|^2 \|w^i\|^2 - (w^T x^i)^2} w_i + \frac{\|w^i\|^2 (y^i - w^T x^i)}{\|x^i\|^2 \|w^i\|^2 - (w^T x^i)^2} x^i \quad (32)$$

This weight update is conditional to the evaluation of a loss function  $l_i$  (defined by (33)) returning a positive value.

$$l_i = \begin{cases} \sum_i y_{pred}^i \neq y^i & \text{for ROMMA,} \\ \sum_i (\max(0, 1 - y^i w^T x^i) > 0) & \text{for aROMMA} \end{cases} \quad (33)$$

In the following section, we extensively evaluate the offline and online learning algorithms for the task of dengue disease detection. As mentioned previously, the baseline results on the dataset is obtained using the offline classifiers. Since the actual system is would need to work on streaming data, the online classifier models are vital in building a robust dengue classification system.

## 4 Experiments and Results

This section describes our experimental setup and results obtained. We first describe the features available in our dataset. Our initial analysis is presented in an offline setting, where we initially use Support Vector Machines (SVMs), Random Forests, as well as multiple classifiers from the Weka Machine Learning library. Subsequently, we evaluate classifiers in an online setting, which establishes the use of our proposed pipeline for detection from streaming data.

### 4.1 Dataset Description

We extracted the following features, which comprise of symptoms and results of pathological investigations, from the dataset of patients in the healthcare information system:

1. No. of days for which symptoms appeared (numeric)
2. Vomiting/Nausea as a symptom (1 if yes, 0 if no)
3. Severe frontal headache (1 if yes, 0 if no)
4. Retro Orbital Pain (1 if yes, 0 if no)
5. Rash (1 if yes, 0 if no)
6. Abdominal Pain (1 if yes, 0 if no)
7. Muscle/Bone Pain (1 if yes, 0 if no)
8. High-grade fever or pyrexia (1 if yes, 0 if no)
9. Hemorrhagic Manifestation (1 if yes, 0 if no)

Table 1: Results using SVM with varying train-test split

Train:Test	Accuracy(%)
10:90	61.34
20:80	65.27
30:70	77.95
40:60	82.24
50:50	78.89
60:40	72.22
70:30	84.90
80:20	<b>85.29</b>
90:10	82.35

Table 2: Results on Random Forest with varying decision trees

#Decision trees	1	2	3	4	5	6	7	8	9
Accuracy	97.92	97.86	97.97	97.64	97.97	98.03	97.97	98.3	97.53
# Decision trees	10	20	30	40	50	60	70	80	90
Accuracy	<b>98.13</b>	87.8	88.08	87.852	87.472	87.638	87.637	87.637	87.967

10. Non-structural glycoprotein-1 (NS1) Antigen
11. Dengue Virus IgM Enzyme-Linked ImmunoSorbent Assay (ELISA)
12. Serum Immunoglobulin IgG ELISA
13. Dengue IgM ELISA 1

The label to be predicted (diagnosis if dengue is positive or not) was represented as the final categorical variable in the dataset. Of the total 182 samples present in the processed dataset, 148 had positive dengue diagnosis and the remaining 34 were diagnosed negative.

#### 4.2 Baseline Results using Offline Classifiers

The first step in developing the model was to ascertain that the feature representation for the data samples was adequate for dengue detection. For this task, we evaluated the classification accuracy on the dataset using SVM classifier for multiple train-test split ratios, as shown in Table 1. It can be seen that though the initial classification accuracy is low, it improves as the classifier model is trained with a larger number of training samples. It is to be noted that the training of the SVM model in this case is not stochastic (or online), rather the entire training dataset is used to train the classifier in one go. One can observe that the highest classification accuracy obtained is **85.29%**, which is obtained when using 80% of the data samples for training and remaining 20% for testing.

The results using random forest classifier with varying the number of decision trees is shown in Table 2. It may be noted that the train-test ratio for these results is kept at 80-20. The highest classification accuracy obtained using random forests for our dataset is **98.13%**.

These results establish that the chosen feature set is suitable for the task of dengue disease classification. The improvement in accuracy when using larger amounts of training data leads us to develop an online learning model that can adaptively improve as more and more labeled data for newer patients becomes available to the dengue detection system. In the present context, we use our dataset with multiple online classifiers to present the viability of the proposed dataset and learning model.

#### 4.3 Results using multiple classifiers

Table 3 shows results on the dataset using various classification algorithms from the Weka library, on the performance metrics of accuracy, true positive and false positive rate, precision, recall, f-measure and Mean Classification Accuracy (MCC). The metrics other than accuracy are often employed in evaluating classifiers on imbalanced data, where the distribution of samples between the classes is skewed. We find that most algorithms perform similarly in terms of accuracy, while random forests outperform other classifiers with an accuracy of 82.41%. Interestingly, the false positive rate in case of random forests is nearly half than that of the other classifiers. This observation is significant in applications such as disease detection where avoiding incorrect diagnosis is of significant.



Table 3: Results using classification algorithms

S.No.	Algorithm	Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
1	Logistic	78.02	0.780	0.753	0.712	0.780	0.735	0.043
2	SGD	81.31	0.813	0.813		0.813		
3	SGD Text	81.31	0.813	0.813		0.813		
4	Simple Logistic	80.21	0.802	0.816	0.660	0.802	0.724	0.051
5	SMO	81.31	0.813	0.813		0.813		
6	Voted Perceptron	81.31	0.813	0.813		0.813		
7	Random Forest	82.41	0.824	0.471	0.813	0.824	0.817	0.382
8	Adaboost	79.67	0.797	0.794	0.699	0.797	0.730	0.006
9	Random Subspace	81.31	0.813	0.813		0.813		
10	Bayes Net	81.31	0.813	0.813		0.813		
11	Naive Bayes	77.47	0.775	0.618	0.751	0.775	0.761	0.177
12	LWL	81.31	0.813	0.813		0.813		

#### 4.4 Results using Online Classifiers

Having established that the generated features from the data can be used to train a learning model for the task of dengue classification, the focus on this section is to evaluate its viability in an online learning setting. To this end, we conduct two sets of experiments, discussed in the following subsections.

##### 4.4.1 Experiments on Incremental Classifier Training

The first set of experiments aims to establish the robustness in training classifiers in an online setting using our dataset. We use the simple perceptron algorithm for this evaluation. We begin with a small sized training set used to train the perceptron model. We incrementally increase the number of samples in the training dataset and compute the error rate, number of updates required for the learning model parameters to converge, and the computational time in training the model. This is done for two cases. In the first case, the classifier model is re-trained with number of samples added incrementally, as well as the samples from which it was trained in the previous iteration. The results in this setting are shown in Table 4.

Table 4: Algorithm :- Perceptron used to examine influence by increasing samples in training dataset on mistake rate, number of updates and execution time by retraining the model using the entire training dataset

S.No.	# Records	Error Rate	# Updates	CPU Time(s)
1	19	0.3684 ± 0.0783	7.00 ± 1.49	0.0038 ± 0.0023
2	37	0.3216 ± 0.0547	11.90 ± 2.02	0.0058 ± 0.0018
3	55	0.3082 ± 0.0331	16.95 ± 1.82	0.0077 ± 0.0021
4	73	0.3027 ± 0.0301	22.10 ± 2.20	0.0083 ± 0.0016
5	91	0.3077 ± 0.0323	28.00 ± 2.94	0.0106 ± 0.0018
6	110	0.3018 ± 0.0271	33.20 ± 2.98	0.0129 ± 0.0031
7	128	0.3066 ± 0.0272	39.25 ± 3.48	0.0157 ± 0.0017
8	146	0.3034 ± 0.0217	44.30 ± 3.16	0.0167 ± 0.0026
9	164	0.2951 ± 0.0237	48.40 ± 3.89	0.0202 ± 0.0029
10	182	0.2981 ± 0.0201	54.25 ± 3.65	0.0221 ± 0.0027

In the second case, the classifier model is trained using only the new samples which are incrementally added at each iteration. These results are shown in Table 5. It can be seen that error rates and number of updates required are comparable in both the cases. The error rate decreases with an increase in training data. It indicates that the classifier model is robustly trained for dengue detection.

One may also note that the CPU time is lower for the case when the classifier model is trained in a purely incremental fashion, i.e., it is re-trained using only the new samples rather than the entire dataset. As the error rates and number of updates are comparable for both the experimental settings, choosing an approach to train the classifier in an incremental manner is beneficial as it saves on computational time, which is critical when the dengue detection system is deployed in a real-time system.

Table 5: Algorithm :- Perceptron used to examine influence by incrementally increasing samples in training dataset on mistake rate, number of updates and execution time by incrementally training the learning model

S.No.	# Records	Error Rate	# Updates	CPU Time (s)
1	19	0.3684 ± 0.0783	7.00 ± 1.49	0.0021 ± 0.0002
2	37	0.3216 ± 0.0547	11.90 ± 2.02	0.0049 ± 0.0015
3	55	0.3082 ± 0.0331	16.95 ± 1.82	0.0072 ± 0.0020
4	73	0.3027 ± 0.0301	22.10 ± 2.20	0.0101 ± 0.0086
5	91	0.3077 ± 0.0323	28.00 ± 2.94	0.0112 ± 0.0014
6	110	0.3018 ± 0.0271	33.20 ± 2.98	0.0126 ± 0.0022
7	128	0.3066 ± 0.0272	39.25 ± 3.48	0.0159 ± 0.0023
8	146	0.3034 ± 0.0217	44.30 ± 3.16	0.0178 ± 0.0018
9	164	0.2951 ± 0.0237	48.40 ± 3.89	0.0192 ± 0.0027
10	182	0.2981 ± 0.0201	54.25 ± 3.65	0.0207 ± 0.0025

#### 4.4.2 Results using various classifier models

The results using multiple online classification models trained incrementally are shown in Table 6. It can be seen that the lowest error rate is obtained for AROW algorithm, while the number of updates required are least for the IELLIP algorithm. This indicates that the generalization of AROW is the best for this dataset, while computational cost is least for IELLIP. The CPU time involved in training these online models is also shown. Though the CPU time is not a bottleneck in the present scenario since the dataset size is limited, the low CPU times indicate that these models are viable for implementation in practical systems where they can be used for real-time dengue disease classification, without heavy computing costs and obtaining predictions within reasonable compute times.

Table 6: Results using online classifiers for dengue detection

S.No.	Algorithm	Error Rate	No. of Updates	CPU Time (s)
1	AROW	<b>0.1901 ± 0.0055</b>	175.45 ± 5.30	0.0411 ± 0.0026
2	OGD	0.1934 ± 0.0088	78.75 ± 3.19	0.0288 ± 0.0015
3	SCW2	0.1973 ± 0.0175	130.90 ± 14.31	0.0355 ± 0.0036
4	NHERD	0.1995 ± 0.0288	175.00 ± 4.65	0.0363 ± 0.0023
5	NAROW	0.2239 ± 0.0709	177.55 ± 5.60	0.0434 ± 0.0010
6	SCW	0.2272 ± 0.1307	65.40 ± 38.90	0.0311 ± 0.0027
7	PA	0.2953 ± 0.0267	93.25 ± 7.68	0.0243 ± 0.0026
8	PA2	0.2953 ± 0.0267	93.90 ± 7.83	0.0266 ± 0.0023
9	PA1	0.2963 ± 0.0267	93.25 ± 7.68	0.0254 ± 0.0019
10	Perceptron	0.2964 ± 0.0271	53.95 ± 4.93	0.0206 ± 0.0031
11	IELLIP	0.2967 ± 0.0266	<b>54.00 ± 4.84</b>	0.0297 ± 0.0018
12	ALMA	0.2981 ± 0.0265	55.75 ± 4.96	0.0253 ± 0.0017
13	CW	0.2989 ± 0.0264	110.35 ± 5.83	0.0326 ± 0.0016
14	SOP	0.3003 ± 0.0200	54.65 ± 3.65	0.0337 ± 0.0022
15	aROMMA	0.0323 ± 0.0449	62.05 ± 8.37	0.0230 ± 0.0018
16	ROMMA	0.3343 ± 0.0481	60.85 ± 8.76	0.0223 ± 0.0021

## 5 Conclusions and Future Work

This paper presented the design architecture and comprehensive evaluation of an online dengue disease prediction model that can be incorporated in a healthcare information system for flagging patients with high probability of being diagnosed with dengue. The model uses simple features which are derived using a combination of recorded symptoms and results of laboratory investigations relevant for diagnosing dengue. The learning model has been trained in an online manner which allows it to incrementally generalize better as more labeled training data is validated by the clinician. Results demonstrate the accuracy and feasibility of the proposed model being deployed in a practical setting for assisting clinicians by providing a data-driven decision model, thereby contributing to overall improvement in healthcare services for disease detection, as well as in identifying trends in varying symptoms for such diseases. Future work involves expanding of the feature space used in this model to include geographic and demographic information, as

well as other parameters such as patient vitals. Similar models can also be developed for detection of other diseases that are of emerging interest to the healthcare practitioners, public health researchers and healthcare policy makers.

## References

- [1] J. A. Potts, R. V. Gibbons, A. L. Rothman, A. Srikiatkachorn, S. J. Thomas, P.-o. Supradish, S. C. Lemon, D. H. Libraty, S. Green, and S. Kalayanarooj, "Prediction of dengue disease severity among pediatric thai patients using early clinical laboratory indicators," *PLoS neglected tropical diseases*, vol. 4, no. 8, p. e769, 2010.
- [2] B. M. Althouse, Y. Y. Ng, and D. A. Cummings, "Prediction of dengue incidence using search query surveillance," *PLoS neglected tropical diseases*, vol. 5, no. 8, p. e1258, 2011.
- [3] E. H. Chan, V. Sahai, C. Conrad, and J. S. Brownstein, "Using web search query data to monitor dengue epidemics: a new model for neglected tropical disease surveillance," *PLoS neglected tropical diseases*, vol. 5, no. 5, p. e1206, 2011.
- [4] N. C. Dom, A. A. Hassan, Z. A. Latif, and R. Ismail, "Generating temporal model using climate variables for the prediction of dengue cases in subang jaya, malaysia," *Asian Pacific journal of tropical disease*, vol. 3, no. 5, pp. 352–361, 2013.
- [5] P. M. Luz, B. V. Mendes, C. T. Codeço, C. J. Struchiner, and A. P. Galvani, "Time series analysis of dengue incidence in rio de janeiro, brazil," *The American journal of tropical medicine and hygiene*, vol. 79, no. 6, pp. 933–939, 2008.
- [6] D. Phung, C. Huang, S. Rutherford, C. Chu, X. Wang, M. Nguyen, N. H. Nguyen, and C. Do Manh, "Identification of the prediction model for dengue incidence in can tho city, a mekong delta area in vietnam," *Acta tropica*, vol. 141, pp. 88–96, 2015.
- [7] C. Rotela, F. Fouque, M. Lamfri, P. Sabatier, V. Introini, M. Zaidenberg, and C. Scavuzzo, "Space–time analysis of the dengue spreading dynamics in the 2004 tartagal outbreak, northern argentina," *Acta tropica*, vol. 103, no. 1, pp. 1–13, 2007.
- [8] S. Bhatt, P. W. Gething, O. J. Brady, J. P. Messina, A. W. Farlow, C. L. Moyes, J. M. Drake, J. S. Brownstein, A. G. Hoen, O. Sankoh *et al.*, "The global distribution and burden of dengue," *Nature*, vol. 496, no. 7446, p. 504, 2013.
- [9] S. Hales, N. De Wet, J. Maindonald, and A. Woodward, "Potential effect of population and climate changes on global distribution of dengue fever: an empirical model," *The Lancet*, vol. 360, no. 9336, pp. 830–834, 2002.
- [10] E. Descloux, M. Mangeas, C. E. Menkes, M. Lengaigne, A. Leroy, T. Tehei, L. Guillaumot, M. Teurlai, A.-C. Gourinat, J. Benzler *et al.*, "Climate-based models for understanding and forecasting dengue epidemics," *PLoS neglected tropical diseases*, vol. 6, no. 2, p. e1470, 2012.
- [11] Y. L. Hii, H. Zhu, N. Ng, L. C. Ng, and J. Rocklöv, "Forecast of dengue incidence using temperature and rainfall," *PLoS neglected tropical diseases*, vol. 6, no. 11, p. e1908, 2012.
- [12] E. Pinto, M. Coelho, L. Oliver, and E. Massad, "The influence of climate variables on dengue in singapore," *International journal of environmental health research*, vol. 21, no. 6, pp. 415–426, 2011.
- [13] A. Earnest, S. Tan, and A. Wilder-Smith, "Meteorological factors and el nino southern oscillation are independently associated with dengue infections," *Epidemiology & Infection*, vol. 140, no. 7, pp. 1244–1251, 2012.
- [14] L. Tanner, M. Schreiber, J. G. Low, A. Ong, T. Tolfvenstam, Y. L. Lai, L. C. Ng, Y. S. Leo, L. T. Puong, S. G. Vasudevan *et al.*, "Decision tree algorithms predict the diagnosis and outcome of dengue fever in the early phase of illness," *PLoS neglected tropical diseases*, vol. 2, no. 3, p. e196, 2008.
- [15] H. M. Aburas, B. G. Cetiner, and M. Sari, "Dengue confirmed-cases prediction: A neural network model," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4256–4260, 2010.
- [16] T. Loshini, V. S. Asirvadam, S. C. Dass, and B. S. Gill, "Predicting localized dengue incidences using ensemble system identification," in *Computer, Control, Informatics and its Applications (IC3INA), 2015 International Conference on*. IEEE, 2015, pp. 6–11.
- [17] T. Faisal, M. N. Taib, and F. Ibrahim, "Adaptive neuro-fuzzy inference system for diagnosis risk in dengue patients," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4483–4495, 2012.
- [18] Y. Wu, G. Lee, X. Fu, and T. Hung, "Detect climatic factors contributing to dengue outbreak based on wavelet, support vector machines and genetic algorithm," 2008.
- [19] Y. Yusof and Z. Mustaffa, "Dengue outbreak prediction: A least squares support vector machines approach," *International Journal of Computer Theory and Engineering*, vol. 3, no. 4, p. 489, 2011.

- [20] S. Khan, R. Ullah, A. Khan, N. Wahab, M. Bilal, and M. Ahmed, "Analysis of dengue infection based on raman spectroscopy and support vector machine (svm)," *Biomedical optics express*, vol. 7, no. 6, pp. 2249–2256, 2016.
- [21] A. L. V. Gomes, L. J. Wee, A. M. Khan, L. H. Gil, E. T. Marques Jr, C. E. Calzavara-Silva, and T. W. Tan, "Classification of dengue fever patients based on gene expression data using support vector machines," *PloS one*, vol. 5, no. 6, p. e11267, 2010.
- [22] A. Radzol, K. Y. Lee, and W. Mansor, "Classification of salivary based ns1 from raman spectroscopy with support vector machine," in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE, 2014, pp. 1835–1838.
- [23] S. Soman, P. Srivastava, and B. Murthy, "Unique health identifier for india: An algorithm and feasibility analysis on patient data," in *E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on*. IEEE, 2015, pp. 250–255.
- [24] S. Srivastava, P. Khurana, A. Rai, A. Cheema, and P. Srivastava, "High performance and adaptive lab report generation in hospital management information systems," in *2016 IEEE Annual India Conference (INDICON)*. IEEE, 2016, pp. 1–6.
- [25] K. Donnelly, "Snomed-ct: The advanced terminology and coding system for ehealth," *Studies in health technology and informatics*, vol. 121, p. 279, 2006.
- [26] C. J. McDonald, S. M. Huff, J. G. Suico, G. Hill, D. Leavelle, R. Aller, A. Forrey, K. Mercer, G. DeMoor, J. Hook *et al.*, "Loinc, a universal standard for identifying laboratory observations: a 5-year update," *Clinical chemistry*, vol. 49, no. 4, pp. 624–633, 2003.
- [27] S. Srivastava, R. Gupta, A. Rai, and A. Cheema, "Electronic health records and cloud based generic medical equipment interface," *arXiv preprint arXiv:1411.1387*, 2014.
- [28] S. Srivastava, S. Soman, A. Rai, A. Cheema, and P. K. Srivastava, "Continuity of care document for hospital management systems: an implementation perspective," in *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance*. ACM, 2017, pp. 339–345.
- [29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [30] T. K. Ho, "Random decision forests," in *Document analysis and recognition, 1995., proceedings of the third international conference on*, vol. 1. IEEE, 1995, pp. 278–282.
- [31] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," in *Advances in neural information processing systems*, 2009, pp. 414–422.
- [32] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 928–936.
- [33] K. Crammer and D. D. Lee, "Learning via gaussian herding," in *Advances in neural information processing systems*, 2010, pp. 451–459.
- [34] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, no. Mar, pp. 551–585, 2006.
- [35] L. Yang, R. Jin, and J. Ye, "Online learning by ellipsoid method," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1153–1160.
- [36] C. Gentile, "A new approximate maximal margin classification algorithm," *Journal of Machine Learning Research*, vol. 2, no. Dec, pp. 213–242, 2001.
- [37] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [38] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "A second-order perceptron algorithm," *SIAM Journal on Computing*, vol. 34, no. 3, pp. 640–668, 2005.
- [39] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," in *Advances in neural information processing systems*, 2000, pp. 498–504.