Application of machine learning to inform clinical management of infectious diseases in Vietnam

Thi Anh Tho Nguyen
B.Eng in Electronic Engineering (Hons)
University of Huddersfield
&
Texas A&M University

Supervised by:
Dr Bernard A Hernandez Perez

A Thesis submitted in fulfilment of requirements for the degree of
Master of Science
Applied Machine Learning
of Imperial College London

Department of Electrical and Electronic Engineering
Imperial College London
September 2, 2021

Abstract

Throughout this project and thesis, a Severe Dengue Predictor was developed and deployed on a web-based application. The model uses LSTM as the basis for the neural network architecture because of its ability to handle time-series data, especially clinical data, which was proven several times through different researches. From the data set provided by OUCRU, a small number of features were selected based on correlation test and WHO Severe Dengue guidelines: *vomiting, body temperature, respiratory rate, haemoglobin, haematocrit percent, platelet count, bleeding vaginal, bleeding mucosal*, and *abdominal pain*.

The neural network architecture has had it hyper-parameter tuned before beginning the training process. The training result was validating through K-Fold cross-validation process, which ensures that biased result is minimal. With models' performance ranging from adequate to excellent, a web-based application has been developed to deploy those train models. A random simulation for the application is also shown in this thesis. Through those simulation, the performance of this machine learning application is verified to be adequate in real-life scenarios.

Acknowledgment

First and foremost, I would like to give my sincerest gratitude to my supervisor, Dr Bernard A Hernandez Perez, for guiding me through this project, and giving me the freedom and support to progress and trial the research to whichever direction I could think of. Every week I received immensely helpful advice that not only applies to this project but will be going along with my machine learning career.

I would also like to show my appreciation to Dr Damien Keng Yen Ming who has given me useful medical advice and helped me gaining experience in working in a group of healthcare technology scientist, which is the direction I hope to start my machine learning career path with.

I want to say the biggest thank to my mother, who has supported me emotionally and financially throughout my educational path. She believes in me more than I do for myself, which gave me the courage to bounce back and keep moving on after the 2018 Egypt bombing incident that has left a deep scar within me. Without my mother I would not be able to become who I am today.

I want to say thank you to my friends and other professors who have made this very difficult Master course very memorable even during a pandemic. And finally other family members who have always given me love and support to go through this year.

Contents

List of Figures

List of Tables

Abbreviations

LSTM: Long Short-Term Memory

RNN: Recurrent Neural Network

RT-PCR: Reverse Transcriptase Polymerase Chain Reaction

DHF: Dengue Haemorrhagic Fever

DART: Daily Risk Tomorrow

DARE: Daily Risk Ever

LASSO: Least Absolute Shrinkage and Selection Operator

AST: aspartate aminotransferase

RMSProp: Root Mean Square Propagation

PPV: Positive Predictive Value

NPV: Negative Predictive Value

Chapter 1

Introduction

This thesis and final year project aims to develop a machine learning predictive model that could aid the healthcare professionals in diagnose, control, and give treatment to Dengue patients efficiently. Using clinical data that has been recorded at Ho Chi Minh Hospital for Tropical Diseases, a Long Short-Term Memory architecture has been developed to diagnose the possibility of developing Severe Dengue complications in Dengue patients.

1.1 Motivation

Machine learning in healthcare is an important application that has been growing rapidly along with big data. Thanks to the healthcare sector that has quickly adapted to the new technology, massive patient's clinical data has been recorded and is easily accessible for data and machine learning scientists to use and develop life-changing technology to aid the process of diagnosis, treatment, etc.

Knowing the power of machine learning, along with the risk and burden of Severe Dengue around the world, there is a lot of interest in using machine learning to diagnose and mitigate the risk for patients, hospitals, and the economy, especially with infectious disease that affects billions of people every year, such as Dengue.

With the seriousness of Severe Dengue, which is a more critical development of Dengue, it is very important that the patient receives proper treatment at the right time to avoid severe complications and even death. However, for developing countries to cope with millions of people during the outbreak period, it is a very devastating scenario where hospitals are out of beds, patients cannot get the much-needed treatment, and where doctors are completely exhausted and overworked. A of people are fortunate enough to get hospitalised. However, a lot of them are also turned out to not develop Severe Dengue, which in the end results in a lot of cost and expense from not just only hospitalisation, medication, …but also loss of work, transportation, etc.

1.2 Thesis Outline

In Chapter 2, background information of Dengue and Severe Dengue will be presented, followed by a brief introduction of the machine learning application that has been researched and developed to control Dengue outbreaks through different means. Chapter 3 will give a more in-depth summary regarding analysis of different types of Dengue classifiers that have been developed throughout the years. Furthermore, two prominent examples of using LSTM in healthcare, which particularly involves using clinical data for training. Chapter 4 introduces the theory side of the machine learning algorithm that will be using throughout this project, recurrent neural network, particularly LSTM, and other neural network components used to optimise performance.

Chapter 5 introduces the data set used for this project, and how 10 variables were identified and used to train the machine learning model. Furthermore, data processing techniques such as data imputation, formatting, standardisation have also been extensively talked about in Chapter 5. After a considerable amount of data information has been covered, a section regarding data visualisation will show readers the nature of clinical data set as a whole, and in particular the OUCRU data set which is used in this project.

Chapter 6 dives into the preliminary neural network architecture along with its hyper-parameter tuning process, including parameters such as dropout rate, LSTM units, learning rate, and number of epochs. The result obtained from training on this architecture is presented in Chapter 7, which also included a section on improving the model through ways of dealing with imbalanced data set. After obtaining different models for Severe Dengue Predictors, an application was built to deploy those models. The application is introduced and a brief instruction on how to use is provided in Chapter 8. Furthermore, Chapter 8 also simulates some known samples to observe how the predictors perform in a more real-life scenario. The observation from simulation is then used in to further improve the initial model in Chapter 9 and get a better performance in simulation.

Some further discussion including weighing the pros and cons of this predictor compared to others in Chapter 3, as well as some recommendations on how to improve the predictor's performance and user experience of the model deployment are discussed in Chapter 10. Finally, Chapter 11 concludes the achievement of this project.

Chapter 2

Background

2.1 Dengue and Severe Dengue

Dengue is a mosquito-borne viral infection that has affected the tropical parts of the world for decades, with hundreds of millions of people affected each year. Since climate changes, global warming, and many other side effects of technology advances, Dengue has started to spread to non-tropical places such as Europe and America.

Dengue virus is transmitted through infected mosquito, primarily *Aedes aegypti*, or *Aedes albopictus* mosquitoes [5]. As of 2015, the *Aedes aegypti* mosquito can be found over 188 countries, posing risks to 3 billion people with contracting Dengue infection [6]. Meanwhile, *Aedes albopictus*, which is considered a secondary vector in Asia, has started to infect American and European countries. It has been identified as the primary source of Dengue infection where *Aedes aegypti* is not present. Compared to *Aedes aegypti*, *Aedes albopictus* is more resistant to climate changes, namely colder conditions, hence its migration evidence is more abundant compared to *Aedes aegypti*. The mosquitoes may have been migrated by chance through international trading of bamboos, tyres, and other goods [5].

Dengue exhibits symptoms similar to flu such as high fever, headache, nausea, rash, ..., and

can affects people of all ages from infant to adult. Most of the time, Dengue will go away after 1-2 weeks without any long-term effect and any hospitalisation. However, in other cases, the infection can further develop into Severe Dengue, which causes complications and has mortality rate up to 20% when untreated [5]. From the first glances, it is hard to determine if a Dengue patient will progress to develop Severe Dengue. From WHO [5], the development of Severe Dengue starts to become diagnosable when the fever subsides. It is during this phase that the combination of different clinical data served as warning signs for Severe Dengue: severe abdominal pain, persistent vomiting, rapid breathing, fatigue, bleeding in gum, and vomit. Some of the outcome of Severe Dengue includes severe plasma leakage (which can then lead to shock), severe bleeding, and severe organ impairment (which affects liver, heart, other organs, and patient's consciousness). Receiving proper treatment on time is extremely important for Severe Dengue, as untreated patients have up to 50% mortality rate while the chances lower to 2%-5% when treated properly [7].

Figure 2.1: Overcrowded Hospital During Dengue Outbreak with 2 Patients Sharing A Bed [1]

Dengue places a very heavy strain on the medical infrastructure and the economy of developing countries. According to Felipe et al, Vietnam is burdened with roughly 2 million infection per year, which costs up to $95 million annually [8]. Most of the cost comes from hospitalisation, loss of income and productivity, transportation, medicines, … [9]. Patient hospitalisation is a big factor when it comes to the burden of the Dengue outbreak. The diagnosis of Severe Dengue is not efficient and has been based mostly on doctor's expertise and experience. During an outbreak, hospitals are overcrowded with patients, doctors are overworked, and their judgement may not be as accurate as the normal time. As a result, at those times, hospital are often overflowing with patients and no available bed. A lot of patients with Severe Dengue may not find a slot to receive proper treatment on time, and a lot more patients may be unnecessarily hospitalised because of misdiagnoses. It is a very serious problem especially in developing countries where hospital infrastructure is not big enough to cope with the outbreak.

2.2 Machine Learning Application for Dengue

Machine learning has been utilised and delivered great models, applications for the healthcare sector. Many diseases can be predicted using machine learning model and health records such as medical images, vital data, test results, etc. A lot of research are focused on using different medical data to diagnose early-stage cancer. Robots are capable of performing surgery with or without assistance, and provide rehabilitation assistance [10].

With many examples that show how useful and effective machine learning is in healthcare, it is inevitable that there is a huge interest in applying machine learning to help control Dengue and reduce the burden it places on the economy. There has been research on using machine learning to forecast Dengue outbreaks using environmental data such as temperature, humidity, wind speed, and rain fall. A particular SVM model from [11] achieved a promising result with a sensitivity of 64%, specificity of 95%, and precision of 56% using those aforementioned features. This result signifies the potential of the project

in controlling a Dengue outbreak. There has been similar research that utilise a wider range of meteorological factors in the analytic model.

The aforementioned types of machine learning models focus on predicting when the next outbreak will happen. It is possible because many meteorological factors such as humidity and temperature can encourage the breeding of *Aedes aegypti* mosquitoes. Successfully predicting the next wave will give the government and suitable authority time to plan ahead their infrastructure and send support to the usual critical area in a timely manner. This is a proactive option to control Dengue.

Oxford University Clinical Research Unit (OUCRU) has taken the lead to carry out studies that provide reactive options to tackle Dengue. In more than 10 years, OUCRU has recruited thousands of patients with Dengue and Severe Dengue to collect medical data. More than 6 different studies are available with their respective data that can be used for early diagnosis and risk prediction models. Traditionally, doctors relied on their expertise, experience, and guidelines from reputable organisations such as WHO to diagnose whether or not a patient may develop Dengue or Severe Dengue. This method is very dependent on the doctors' skills and in an outbreak settings, manpower can be limited. This is why reactive as well as proactive control are both important in controlling Dengue.

With the database from OUCRU, several research has been done in using statistical models to analyse the data and find out the important features which lead to the diagnosis of Dengue and Severe Dengue [12] [13] [14]. Furthermore, several classifiers have been developed such as the Early Dengue Classifier with a sensitivity of 74.8% and specificity of 76.3% [15], which will be further analysed in Chapter 3.

Chapter 3

Literature Review

3.1 Early Dengue Classifier

To aid the fast and early identification of a Dengue outbreak, medical doctors and researchers were interested in using NS1 Ag Strip rapid test. The gold standard of Dengue testing comprises RT-PCR, IgM serology, and NS1 ELISA. According to [16], RT-PCR bears a sensitivity and specificity of 89.9% and 100%, respectively. NS1 ELISA performance scores between 76%-97% for sensitivity and 98%-100% for specificity as stated in published literature. Compared to those gold standard tests, NS1 rapid test bears a lower performance of sensitivity and specificity of 81.5% and 66.7% according to present study [16].

With the NS1 rapid test having lower performance, it was in the researchers' goal to develop an Early Dengue Classifier to use in conjunction with the rapid test to increase overall performance. A multivariate logistic regression model with clinical data from OUCRU was developed to detect Dengue within the first 72 hours of illness. The optimal set of features comprises age, white cell count, and platelet count. It is necessary to have as few features as possible and still perform adequately to allow a speedy and reliable diagnostic. [15]

The Early Dengue Classifier has a sensitivity and specificity of 75% and 76% by itself. When using in conjunction with the NS1 Ag rapid test, the combined sensitivity and specificity increase to 91.6% and 75.7%. A higher specificity can also be achieved by using a higher cut-off value. For example, with a cut-off value of 50% (previously 33.3%), the Early Dengue Classifier specificity can rises to 89.6% while sensitivity lowers to 86%. [15]

The development of Early Dengue Classifier shows that the use of machine learning in both reactive and proactive control is realisable and will bring great results when using in conjunction with pre-existing medical technology. The authors of Early Dengue Classifier, Nguyen Minh Tuan et al [15, p. 10] emphasises the strength of their study based on "large sample size, the presence of all four DENV serotypes, robust statistical validation techniques and transparent performance characteristics". Based on the performance of Early Dengue Classifier, other researchers have an evidence-based simple, but working model to start with and further improve its performance through other statistical and machine learning methods. One of the cons of the Early Dengue Classifier is its robustness. The classifier relies on hematology findings that are not accessible outside of primary care, and the use of age in features will likely reduce the model's generalisation power in other populations where Dengue affects other age groups differently. Despite the weaknesses presented above, it has paved the way in developing a predictor to aid the doctors and hospitals in Dengue diagnosis, patient triage, and outbreak management.

3.2 Dynamic Dengue Haemorrhagic Fever Calculators

Dengue Haemorrhagic Fever (DHF) is one of the Severe Dengue outcomes defined by the World Health Organization. In 2019, 2 different calculators were developed to calculate the risk of developing Dengue Haemorrhagic Fever from Dengue patients. The features for 2 calculators are selected using LASSO. There are 2 different types of features used in the research conducted by Ken Wei Tan et al [4], daily features and admission features. Daily features are daily clinical measurements that may change drastically throughout a patient's hospitalisation. And admission features are measurements collected at the

admission of patients to the hospital, which will be constant.

The first calculator DART (DAily Risk Tomorrow) calculates the risk of having DHF the subsequent day based on daily clinical measurements and some other clinical data at admission. Some of the very important features found for DART are platelet counts, maximum temperature on day 6, maximum basophil count, minimum diastolic blood pressure, and number of day(s) since fever onset. Those features have high coefficient in DART model, which helps adding great predictive power to the calculator. DART scores 97.6% and 9.4% for sensitivity and specificity, respectively. While the specificity is low, the sensitivity score is very promising as it demonstrates the ability of the calculator to avoid false-negative diagnosis of patients with DHF. However, if the calculator is intended to use to reduce the burden on the hospital system, the model needs to be further improved, especially on its specificity score [4]. This remark is further supported by Table 3.1 where DART's PPV and NPV are recorded to be 1.9% and 99.5 % respectively. Those figures mean that out of everyone that is tested positive by DART, only 1.9% turns out to be true positive. With this performance, DART will not be able to adequately reduce the number of patients overflowing the hospitals during an outbreak and help reduce the burden on hospitals and the economy.

	Sensitivity (%)	**Specificity (%)**	**PPV (%)**	**NPV (%)**
DART	97.6	9.4	1.9	99.5
DARE 1-4	97.6	15.4	9.7	98.6
DARE 5	97.8	16.1	9.2	98.8
DARE 6	97.6	16.2	7.3	99
DARE 7	97.8	15.7	4.8	99.4
DARE 8	98.4	24.9	2.6	99.9

Table 3.1: Performance of DART and DARE Calculators [4]

The second type of calculator is DARE (DAily Risk Ever), which predicts the risk of having DHF at a day in the future using only admission features. In research done by Ken Wei Tan et al. [4], 5 different DARE models were developed, which predict 1-4, 5, 6, 7, or 8 days into the future if the patient may face Dengue Haemorrhagic Fever on

that day. The performance of DARE calculators is shown in Table 3.1. As the calculator predict further into the future, its performance decreases in term of PPV, while NPV steadily increases.

Through the research, some variables were suggested to have a positive association with DHF such as pre-existing lung or cardiac conditions, and HIV/AIDS. Other characteristics such as being younger, male, and ethnic Indian are found to be negatively associated with DHF. Furthermore, the warning signs in WHO guidelines are observable in this study.

In this study, it is mentioned that one of the biggest "hindrances to the generalisability of DHF calculators …is the number of measurements required" [4, p.13]. Just like a usual data set, there are missing values amongst this data. Although all features are recorded at admission, not all of them are updated throughout a patient's stay. This problem was solved with imputation and some degree of assumption. However, there is still a problem if trying to deploy this model and run a clinical trial. A lot of data recorded from medical tests are not feasible to get recorded every day. As a result, for real-life application, missing values will be a recurrent problem.

3.3 Prognostic Models for Early Identification of Severe Dengue Cases

Early risk prediction of Severe Dengue is crucial in hospital systems and outbreak management for endemic countries. An Early Severe Dengue Identifier was developed by Nguyen Minh Tuan et al. [17] using logistic regression with features that were verified to be independently associated with Severe Dengue through the multivariate logistic model. The features of the final model include history of vomiting, platelet count, AST, and NS1 rapid test status at the time of enrollment. This is a simple model, however, the result shows how promising this model can be with sensitivity and specificity of 87% and 88%

respectively.

Compared to the DART and DARE calculators where more features were used in models, the performance of ESDI is surprisingly good. ESDI uses some features that are associate with WHO warning signs such as vomiting and platelet count. The performance of ESDI suggests that the incorporation of features related to Severe Dengue may enhance the performance of the predictor.

ESDI is supposed to be able to predict all the outcomes of Severe Dengue, which includes severe plasma leakage, severe haemorrhage, severe organ impairment, and shock. However, in the actual paper, Nguyen Minh Tuan et al. [17] did not clarify the distribution of those outcomes within the data set. From both of the research papers [17] [4], some very important features have been identified will be referred to when developing a new predictor for this project.

3.4 LSTM in Healthcare

LSTM is a type of recurrent neural network, which will be explained in more detail in Section 4.2. LSTM's specialty is handling time series data, which is most of the clinical and biomedical data. As a result, there is great interest in using LSTM for the healthcare machine learning model.

Jing Xia et al. [2] finds the necessity of having a model predicting ICU patients' mortality as well as a model improving doctor's prognosis accuracy. LSTM was compared against several other algorithms (SAPS II, SOFA, APACHE II) and was found to have the best performance when using clinical data to assess patient's risk of death.

In Figure 3.1, it can be observed that with only 3 days worth of data, LSTM and eLSTM

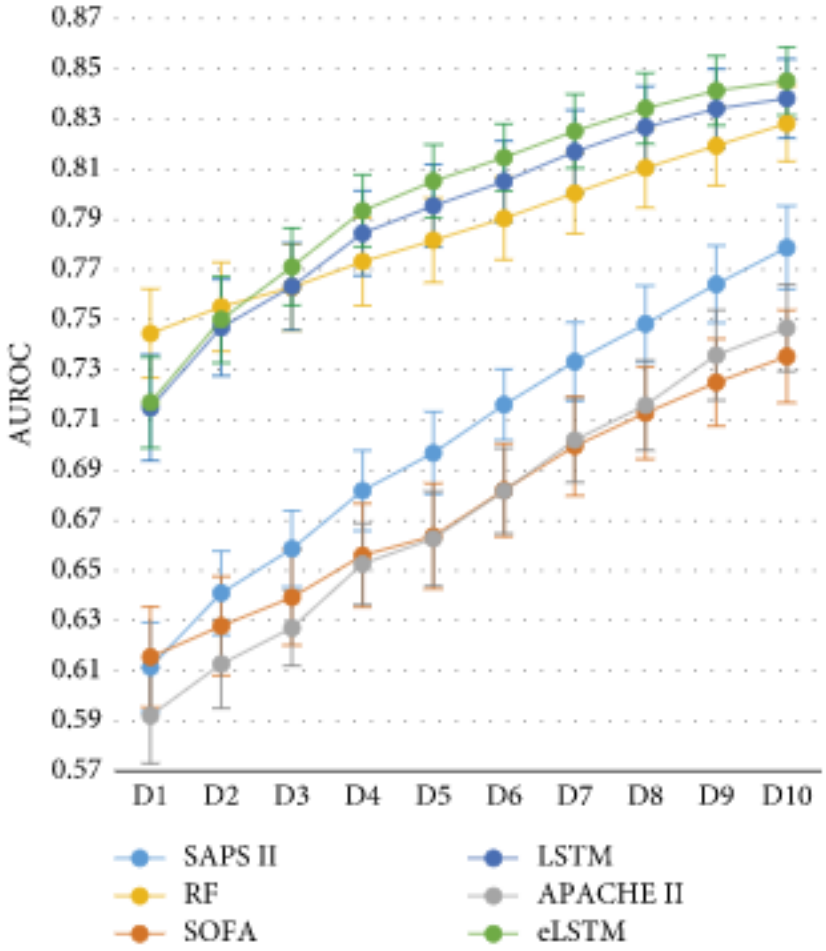


Figure 3.1: Performance of Mortality Predictors Day 1-10 [2]

were able to reach the highest performance that SAPS II, SOFA, and APACHE II models obtained on day 10. For an ICU patient, a precise and early mortality prediction is essential for the doctors to devise new treatment plans and for hospitals to give patients the proper priorities. Because of its ability to observe the trend and dependencies in clinical data throughout time, LSTM was able to pick up more learning hypotheses than a normal model that cannot handle temporal data.

Another application that has found success in LSTM is predicting healthcare trajectories from medical records. Modern electronic medical records have partially eliminated the doctor's need to ask for a brief medical history recap from a patient. The diagnosis

accuracy depends not only on the doctor's experience and expertise, but also on the patient's ability to recount his or her medical history accurately. Using LSTM-based architecture, Trang Pham et al. [18] developed DeepCare, a neural network model that reads electronic medical records, takes current illness state, data and predicts medical outcomes for the future. Electronic medical records contain data such as family medical history, past diagnosis, symptoms, clinical data from monitoring devices, and clinician's observation. In the research [18], DeepCare was tested to predict unplanned readmission of diabetes and mental-heal patients within 12 months and 3 months, respectively. With the best F-score of 79%, DeepCare has proven its feasibility of using LSTM and big medical data to predict health trajectories up to a year in advance.

From the report of LSTM performance on healthcare data, it is quite reasonable to make the assumption that an LSTM-based architecture for Severe Dengue Predictor would produce a model with better performance than models using logistic regression and multivariate logistic because of clinical data's time series nature. The memory of a recurrent network allows the model to process, learn, and memorise the knowledge taken from a sequence of data.

Chapter 4

Machine Learning Algorithm

4.1 Recurrent Neural Network

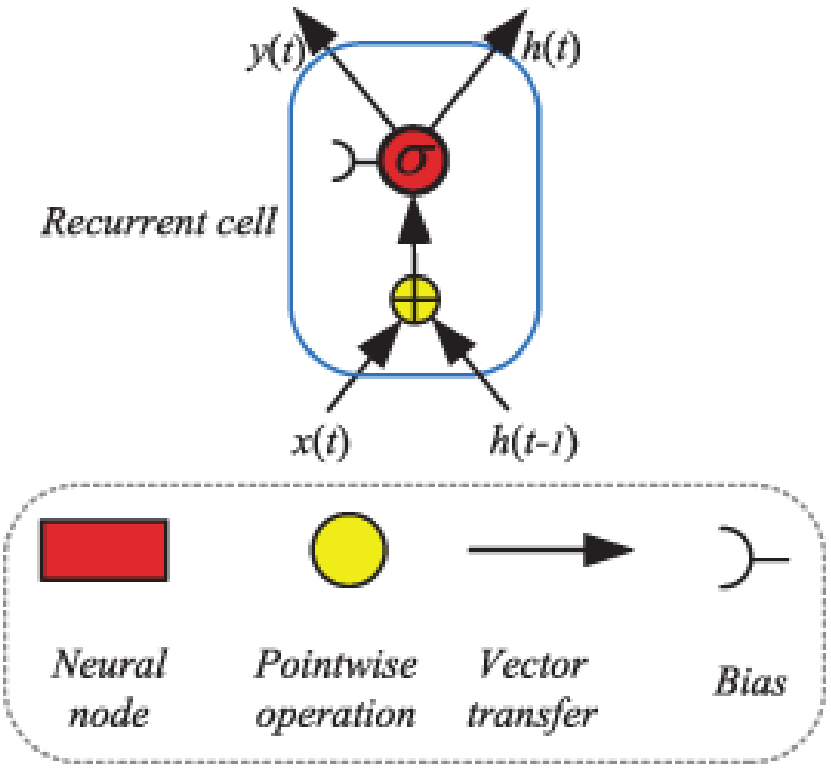


Figure 4.1: Schematic of RNN cell [3]

RNN is a type of neural network that allows the use of previous output as the next state

input. It has the advantage of taking account of historical data, which is why it is popular for time series data [19]. A RNN typically contains several recurrent cells using tanh or sigmoid activation functions. The cell from Figure 4.2 can be expressed mathematically:

$$\begin{aligned} h_t &= \sigma(W_h h_{t-1} + W_x x_t + b), \\ y_t &= h_t \end{aligned} \tag{4.1}$$

Where x_t, h_t, and y_t represent the input, hidden state or recurrent information, and the output at time t, while W is the weight for the respective element. Compared to a regular fully-connected layer, RNN has a hidden state element in its system that takes into account the previous state input's representation. This element allows RNN architecture to process a long sequence of data one-by-one, with previous information being passed on to the next input. If a time-series sequence is to be fed into a traditional feed-forward neural network, all of the data will be fed in at the same time, and there will be dependencies between time-step that a traditional neural network cannot pick up.

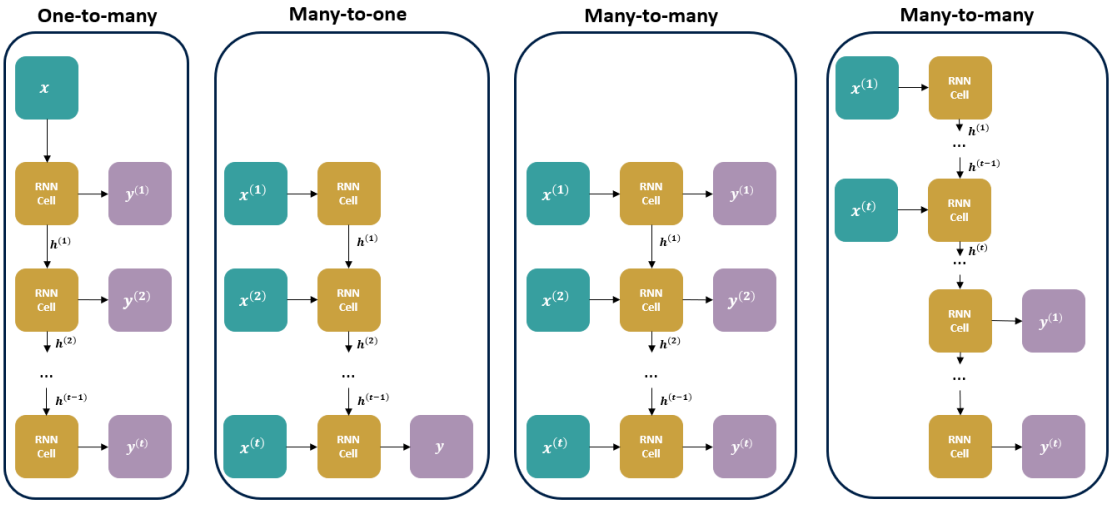


Figure 4.2: Types of RNN Architectures

Because a sequence of data is fed one-by-one into RNN, there can be many types of RNN architecture such as one-to-many, many-to-one, and many-to-many. Each type

of architecture has a unique combination between input and output. The one-to-many type takes a single observation (image or word,…) and has many RNN cells to produce multiple outputs at each cell. One example of one-to-many RNN type is music generation where a single musical note or tone is fed into the model and multiple notes will be produced in sequence. In this case, the previous resulted musical note will affect the next one and so on so forth. The many-to-one architecture takes a sequence of observation and performs classification, which is usually seen in sentiment classification problem. The many-to-many architecture finds its application in many problems such as machine translation, generating text, speech recognition, …

For RNN, the final loss $\mathcal{L}$ is an accumulation of each loss at each time-step. Thus, the loss optimisation process is done at each time-step T, which can be expressed as shown in Equation 4.2, where the gradient at time-step T is calculated by taking the derivative of $\mathcal{L}$ at time-step T with respect to the weight matrix $\mathbf{W}$.

$$\begin{aligned} \mathcal{L}(\hat{y}, y) &= \sum_{t=1}^{T_y} \mathcal{L}\left(\hat{y}^{(t)}, y^{(t)}\right) \\ \frac{\partial \mathcal{L}^{(T)}}{\partial W} &= \sum_{t=1}^{T} \frac{\partial \mathcal{L}^{(T)}}{\partial W}\bigg|_{(t)} \end{aligned} \tag{4.2}$$

RNN suffers from short-term memory because of the vanishing gradient problem. Vanishing gradient is when the updated value after back-propagation becomes so small that it does not contribute anything to the already learned information. Eventually, the layers that get these small update values stop learning. In RNN, the vanishing gradient usually occurs at the first layers in a longer sequence. As a result, the model may forget, or leave out important details at the beginning of the sequence [20].

4.2 Long Short-Term Memory

LSTM is an improved version of the traditional recurrent neural network developed to solve the short-term memory or vanishing gradient problems in back-propagation by having multiple gates to control the learning of a cell, including forget, input, and output gates. Furthermore, aside from the hidden state which carries the previous state's representation, a cell state is also added into LSTM. Compare to a hidden state, a cell state has better long-term memory capability, which can store not only information from the immediately previous state, but also many other previous states.

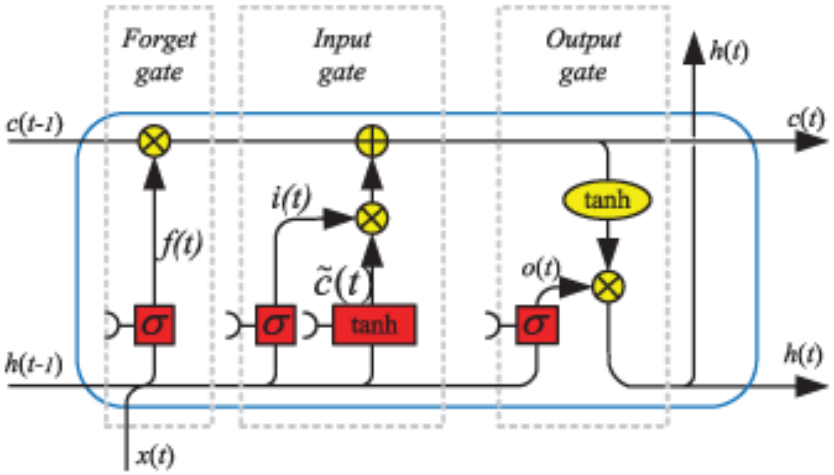


Figure 4.3: Schematic of LSTM Cell [3]

A forget gate f_t determines which information in an input sequence is not important and can be ignored, essentially allows LSTM to only pick the relevant information to learn. An input gate i_t and $\tilde{c}_t$ is used to update the cell state using previous hidden state and current input. Both of the inputs are passed into the sigmoid and tanh activation function in the input gate. By passing them into the sigmoid function, it does the exact thing as the forget gate does, choosing which values to be important and not. Meanwhile, passing the inputs into the tanh function will transform the values to be between $[-1, 1]$ and helps them not exploding when going through mathematical operations. The outputs of for-

get gate, $f_t \cdot c_{t-1}$, and input gate, f_t, $i_t \cdot \tilde{c}_t$ are added together and become the new cell state.

$$
\begin{aligned}
f_t &= \sigma\left(W_{f(h)} h_{t_1} + W_{f(x)} x_t + b_f\right) \\
i_t &= \sigma(W_{i(h)} h_{t-1} + W_{i(x)} x_t + b_i) \\
\tilde{c}_t &= tanh\left(W_{\tilde{c}(h)} h_{t-1} + W_{\tilde{c}(x)} x_t + b_{\tilde{c}}\right) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\
o_t &= \sigma\left(W_{o(h)} h_{t-1} + W_{o(x)} x_t + b_o\right) \\
h_t &= o_t \cdot tanh(c_t)
\end{aligned}
\tag{4.3}
$$

Alternatively, we also can observe how LSTM works through Equation 4.3. All the forget gate and input gate, or calculation of f_t, i_t, $\tilde{c}_t$ is to get the cell state c_t. Equations 4.3 and Figure 4.3 demonstrate that a cell state only carries relevant information from the last states as well as the current state. The cell state is then used to calculate LSTM cell output h_t and transferred to the next cell. The addition of cell state allows LSTM to easily process the long sequences that often pose problems to RNN [3]

4.3 Dropout

Dropout is a regularising method effective at reducing the overfitting problem usually presented in machine learning models. While there are other methods to reduce overfitting, they are often complicated and requires more computational power. Dropout is very popular in deep learning because of its cheap computational expense.

Depends on the size of the model, a neural network architecture can consist of hundreds to millions of nodes or activation functions. When training a large network on a relatively smaller data set, overfitting will undoubtedly become a problem which signals that the generalisation power of the model is poor. This happens because the model is so complex,

it was able to "memorise" everything it sees from the small training set, which in turn gives very little space to interpret unseen data. Using dropout, a portion of random nodes will be deactivated at each turn, which forces a sparse representation of node's learning and at the same time forces some other nodes to take on more responsibilities (to correct other nodes' losses). This algorithm creates a more robust model when it comes to unseen data, which is the entire point of machine learning.

4.4 Optimizer

When a sample finishes propagating through the network and produces a prediction $\hat{y}$, the optimizer calculates the loss $\mathcal{L}$ and uses $\mathcal{L}$ to optimise learning (reduce loss) by updating the weight using a specific algorithm.

Gradient Descent is the very basic but most popular algorithm for optimisation. By taking the derivative of the loss function, the algorithm finds the direction towards the minimum point and updates the weight towards that point. An important hyper-parameter of Gradient Descent is learning rate η, which determines how much the weight would change. A small η may require a lot of computation while a larger one may make the algorithm completely miss the optimal point. Gradient Descent only updates weights after a whole data set has been fed to the model (1 epoch). As a result, if the data set is too large, calculating the gradient of the whole data set will take a long time and the algorithm may never converge. Furthermore, because the gradient is calculated only once per epoch on the whole data set. To solve the problems of Gradient Descent, a new variant called Mini-Batch Gradient Descent is developed in which the weights are updated after every batch. With all Gradient Descent variants, there are common challenges which include getting trapped at local minima (gradient at local minima is still 0, thus not updating the weight), and having constant learning rate η.

Root Mean Squared Propagation (RMSProp) is a Gradient Descent based optimisation

algorithm that provides solutions to the Gradient Descent variants' problems. RMSProp uses adaptive learning rate through moving average. As the value of moving average increases, the learning rate becomes smaller which allows the algorithm to converge. The algorithm is described in Equation 4.4, where $E[g^2]$ is the moving average of the squared gradient, η is the learning rate, and γ is the moving average parameter, or discounting factor for the history/coming gradient. The current moving average is calculated based on a fraction of previous moving average of squared gradients and the remaining fraction of the current gradient. Hinton, the author of this optimizer algorithm suggests that the optimal value for γ and η are 0.9 and 0.001, respectively.

$$
\begin{aligned}
g &= \frac{1}{m}\sum_{1}^{m}\mathcal{L}(\hat{y}, y) \\
E[g^2]_t &= \gamma E[g^2]_{t-1} + (1-\gamma)g_t^2 \\
w_{ij}(t) &= w_{ij}(t-1) - \frac{\eta}{\sqrt{E[g^2]+\epsilon}}g_t
\end{aligned}
\tag{4.4}
$$

RMSProp is a suitable optimizer for the proposed neural network as it uses LSTM, which is considered more complex compared to a regular Dense layer because of its architecture. The complexity of the proposed neural network makes it susceptible to counter the exploding or vanishing gradient problem. Those problems often arise when using a fixed learning rate because each weight may require different learning rates because they have different weights. RMSProp solves the problem by adjusting the learning rate using the previous moving average of the squared gradient.

Overall, the preliminary neural network design is very simple. The reason is that within the data, there a large amount of 0 values, which the model will eventually learn that it is just padding. Thus, the data may not be sufficient to be used in a complex architecture so the training and evaluation will start with a preliminary, simple design, and move on to another more complex one if needed.

Chapter 5

Data Processing

5.1 Overview of Data Set

The data used for this research is "Inpatient-based study examining prognostic factors during the febrile phase", a project of OUCRU where effort is made in using big data and machine intelligence to control Dengue. Patients from 5-15 years old admitted to Dengue ward at Ho Chi Minh Hospital for Tropical Diseases were recruited for the data collection. There are a total of 2,615 participants with confirmed Dengue in the completed data set. Dengue is confirmed by a reverse transcriptase-polymerase chain reaction (RT-PCR) test or IgM & IgG test.

Patients that were diagnosed to may have Severe Dengue were transferred to the Paediatric Intensive Care Unit (PICU) and further be examined and monitored for the study. Because patients are recruited since the febrile phase, a lot of Severe Dengue patients presented in this data set come with weeks worth of data. This makes the data set very informative and promising to use for the training of Severe Dengue Predictor.

5.2 Feature Selection

The data set from OUCRU originally has 70 clinical features, to begin with. However, after inspection at the number of NaN values, a few features have been deemed poorly recorded amongst patients. The inclusion of such features may bring outliers to the data. Thus, 18 poorly-recorded features were permanently excluded from this project.

Correlation between features and outcomes has been calculated using Pearson's coefficient. Pearson coefficient is able to sufficiently calculate the correlation between dichotomous (binary) variables. However, depends on the data, the result may not be as informative as those of other coefficients, such as the Phi coefficient, Carmer's V, Tschuprow's T, Contingency coefficient C, which all uses Chi-squared statistic, a popular approach for dichotomous variables. Figure 5.1 shows the correlation using Pearson's coefficient, which will be used as a reference for feature selection.

From Chapter 3, it was observed that using variables related to the warning signs may have positive effects on the performance of models. As a result, the training data will include mostly variables chosen from WHO's guidelines. According to 2009 WHO guidelines [21], Dengue warning signs consist of *abdominal pain, persistent vomiting, clinical fluid accumulation, mucosal bleeding, lethargy* or *restlessness, liver enlargement*, and behavior over time of *haematocrit ratio*, and *platelet count*. Some of the warning signs are available in the data set even after the purge of poorly-recorded features, such as *abdominal pain, vomiting, mucosal bleeding, haematocrit ratio* and *platelet count*. On the other hand, the *restlessness* feature was very poorly recorded and could not be included in the model. Furthermore, the data set does not lend itself to look at *clinical fluid accumulation* easily. However, Deborah HL Ng et al. [22] have suggested that there is an association between *clinical fluid accumulation* and *saddleback fever*. Moreover, prolonged fever may be associated with Severe Dengue, according to [22]. As a result, *body temperature* feature was included in the feature group despite not having a high correlation with Severe Dengue outcome when looking at Figure 5.1. Another potential

feature to look at is *respiratory rate*. According to WHO 2021 information on Dengue [5], rapid breathing is also a warning sign of Severe Dengue.

When looking at Figure 5.1, some of the aforementioned warning signs have an adequate correlation with Severe Dengue outcomes aside from *respiratory_rate* (`respiratory_rate`, *platelet count* (`plt`). This may be the result of the clinical features being dependent on each other when diagnosing Dengue outcomes. For example, as stated by WHO [5], one of the warning signs includes the increase of haematocrit ratio *concurrent* with a rapid decrease in platelet count. Since `bleeding_vaginal` presents quite a noticeable correlation with `sd_bleeding`, it is also included in the model.

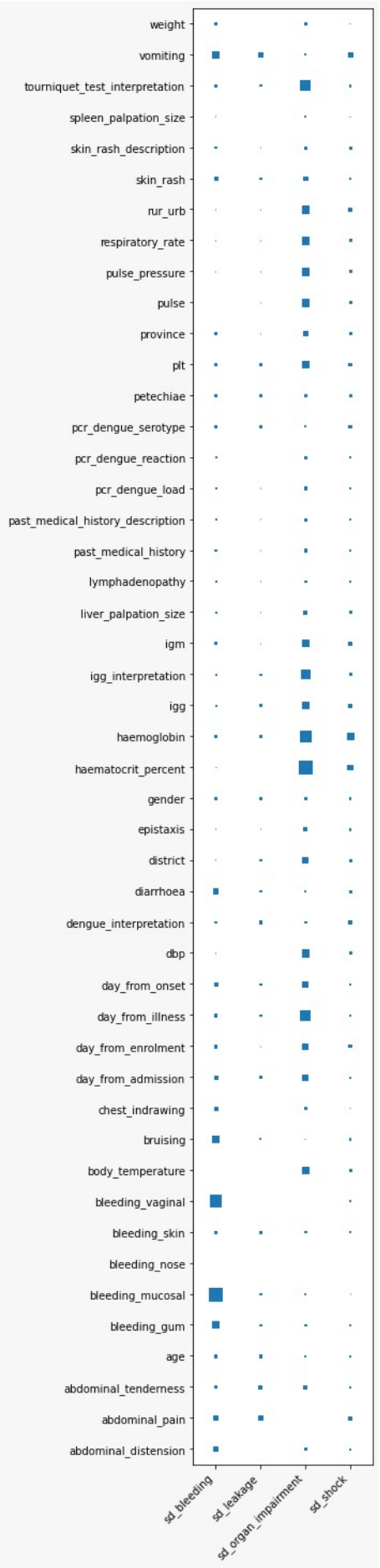


Figure 5.1: Correlation of Features and Outcomes

5.3 Data Imputation

OUCRU data set contains a lot of missing or `NaN` values. It is reasonable to expect `NaN` values more often in clinical data set than regular data set, since a lot of comprehensive tests on patients are not done on a daily basis for cost and efficiency purposes. As the amount of `NaN` values is high, using average values data imputation will not work well.

According to François Chollet [23], replacing `NaN` values with 0 should not affect the neural network's performance in the condition that 0 is not a meaningful value. Amongst all the numeric clinical data in the chosen features, there is not any that uses 0 to represent meaningful data. For binary features, `NaN` or `False` usually both mean that there was not any noticeable patient's reaction that could have been observed. For example, the *vomiting* feature with `False` annotation means that the patient did not vomit. And it is reasonable to assume that with `NaN` values in *Vomiting*, there was also no vomiting observed from the patient.

As a result, all missing values are imputed with 0 for this project. This also simulates the reality that not all tests are done daily and in fact, some tests are only done when the doctor decide that they are necessary

5.4 Data Formatting

The proposed model for this project is LSTM, which requires a more specific shape of data when feeding into the model for training than the usual model. The shape requires by LSTM is `(time_step, feature)`, as shown in Figure 5.2. `time_step` represents the length of a sequence (or a sample). And `feature` tells the number of features included in a sequence. In this particular project, `time_step` means the number of days with a patient's clinical data to be used for training and prediction. As a result, `time_step` varies between patients and also depends on the number of days to be predicting ahead

of.

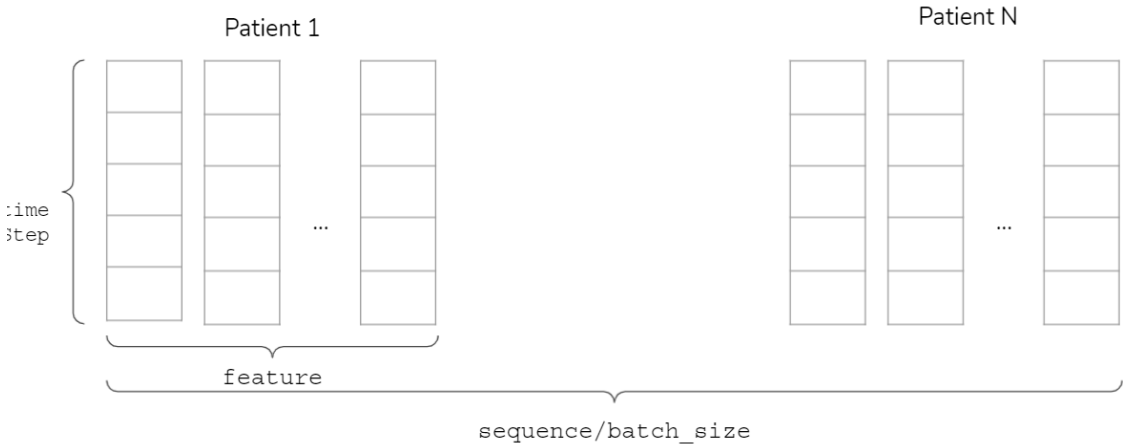


Figure 5.2: Overall Visualisation of LSTM Data Shape

Figure 5.3 shows a more in-depth visualisation of the formatted input and the associate outcome. The main goal of Severe Dengue Predictor is to determine if a patient will develop Severe Dengue in the next N day(s). Any patient with original `time_step` less than N will be discarded entirely. While other sequence/patient will have N last rows of data taken away. For example, if patient 2405 originally have 4 days worth of data (original `time_step` of 4), and needs to be formatted appropriately to train a 2-day-ahead predictor, the last 2 days (or rows) of data will be discarded. And the outcome of this sequence depends on if patient 2405 developed Severe Dengue on day 4 or not. Thus, we will use clinical data up to day 2 to predict the outcome at day 4, for a 2-day-ahead predictor.

It is inevitable that there will be different `time_step` within the data set, just as each patient would have been hospitalised different lengths of time, and had their clinical record done at different levels. Because the data shape at the first layer is `(time_step, feature)`, it requires the same `(time_step, feature)` in one batch, which would therefore require having multiple set of inputs with different `time_step`. One way to avoid that is to find the longest `time_step` and pad shorter samples with 0 value across

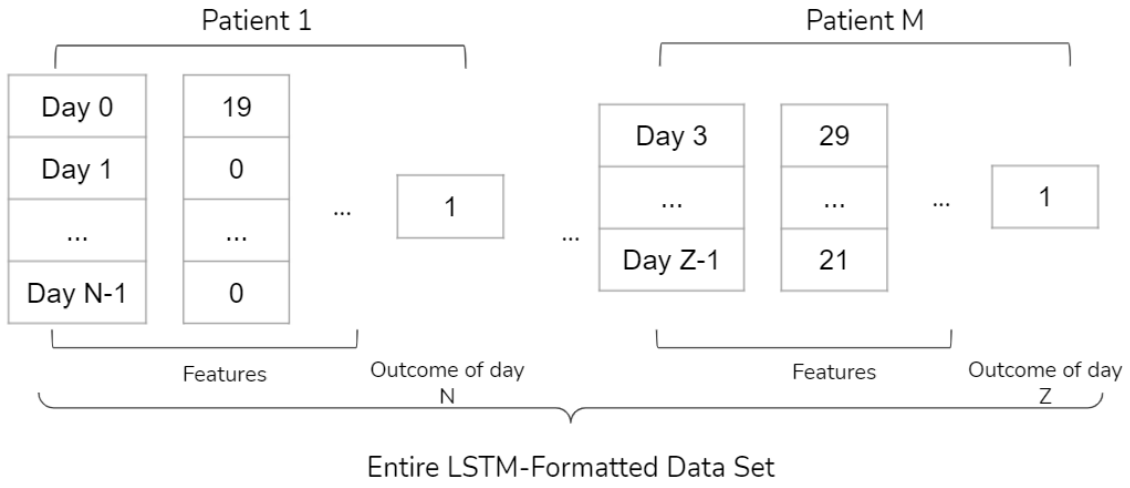


Figure 5.3: Visualisation of LSTM Data Shape and Its Output for 1-Day-Ahead Prediction

all features. Padding is a common practice when using any type of RNNs network and/or time series data to work around sequences with variable lengths, such as text sequences.

Overall, the algorithm to format data appropriate for LSTM can be summarised as follow:

1. Select features from the entire data set
2. Join data and outcome: for each patient,
    - if he/she did not develop Severe Dengue:
        (a) Join all the rows of his/her data except for the last day
        (b) Record the outcome as negative for the current patient
    - if he/she developed Severe Dengue:
        (a) Join the rows of his/her data until 1 row before the Severe Dengue outcome is confirmed
        (b) Record the outcome as positive for the current patient
3. Step-ahead format: for each patient,
    - If time-step is equal to 1, skip the step-ahead format

- If data length of the current patient is less than time-step (does not have sufficient data to perform prediction X-step-ahead), delete the patient's data entirely
- If data length (rows of data) of the current patient is greater than time-step, drop the $X - 1$ last rows of current patient's data

4. Remove patients that contain only 0 or `NaN` in their features
5. Decide on a common data length
   - If current patient's data length is less than common data length, pad with rows of 0 to achieve common data length
   - If current patient's data length is more than common data length, delete the patient's data entirely

5.5 Further Preprocessing and Evaluation Procedures

The data set has been cleaned up previously by OUCRU so it does not need a lot of preprocessing except standardisation. A `RobustScaler` by Scitkit-learn was used for standardisation. `RobustScaler` standardises data by using the 1^{st} and 3^{rd} quartile. Median and interquartile ranges from the training data can be stored and used on testing data or individual samples when deploying the model.

For performance evaluation of the model, the K-fold cross-validation technique is used, where the procedure is as follow:

1. Shuffle the entire data set
2. Split the data set into K groups
3. Rotate between the K groups :
   (a) Take 1 group for the testing set

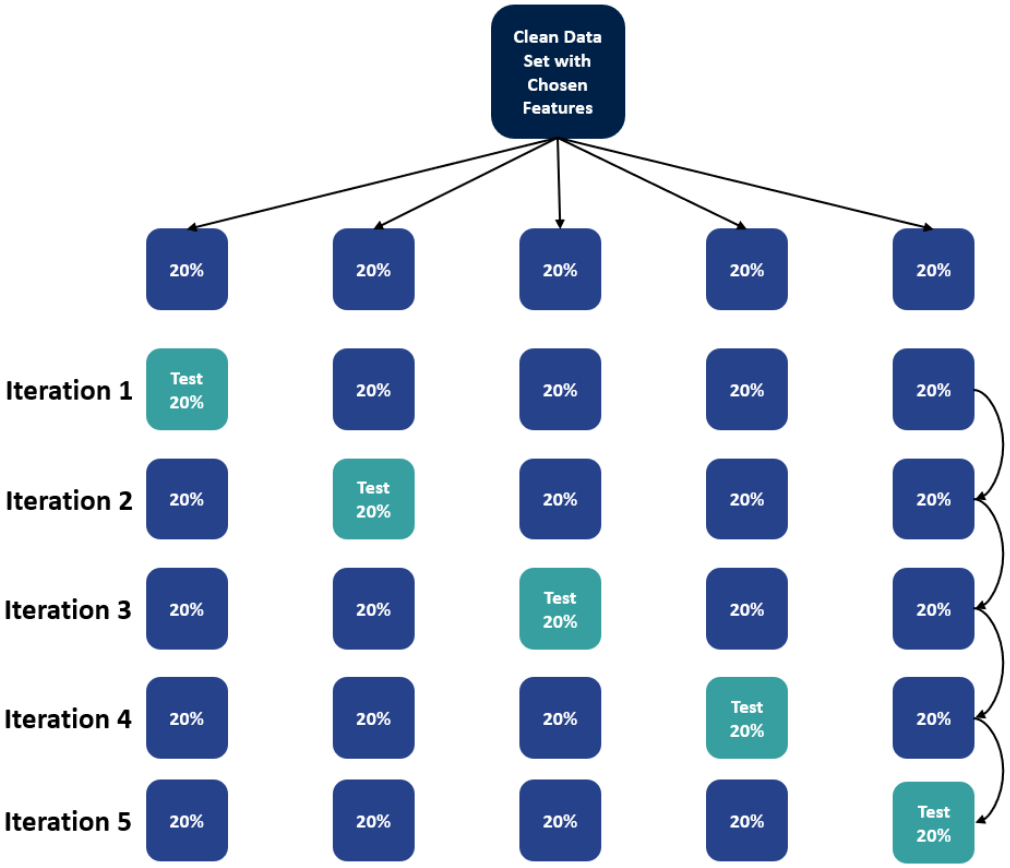


Figure 5.4: K-Fold Cross-Validation Visualisation

(b) Take the remaining $(K-1)$ group for the training set

(c) Fit and train the model on the training set

(d) Evaluate testing set

(e) Retain the evaluation score and discard the model

4. Summarize the performance by finding the average evaluation scores of all iterations

Specifically, the evaluation is done on 5-fold cross-validation, which means a standard 20% of the data set is used for evaluation. The reason why K-fold cross-validation is popular for model evaluation is because it ensures that every data has a chance to appear in the training and testing process, which gives a less biased result overall.

5.6 Data Visualisation

5.6.1 Outcome Distribution

Even though the study recruited a large number of patients, and a decent number of patients turned out to develop Severe Dengue, there is no trace of 2 out of 4 Severe Dengue outcome as shown in Figure 5.5, *severe leakage* and *severe bleeding*. Upon inspection, it was discovered that 1 patient was recorded with *severe leakage* outcome. Unfortunately, according to the data, the patient seems to have already had developed *severe leakage* prior to admission. Similarly, there were 2 patients admitted to the study already having *severe leakage* condition. Thus, for these patients, after the data went through the formatting phase, there was no trace of those 2 Severe Dengue outcomes.

Initially, the binary distribution started out on the more positive side. At 1-step-ahead format, 59% of the sample is positive. However, as the format changes to 2-step-ahead, the positive data decreases to 54%. It means that from in the initial format, 1-step-ahead format, there were some patients that have only 1 row/day worth of data to process. As a result, when moving to the 2-step-ahead format, having to drop another 1 row, those patients no longer exist in the data set, which reduces the number of positive samples, and its distribution. The same thing happens over and over again as the step-ahead format increases. At 9-step-ahead format, the entire `sd_shock` outcome is no longer available, and the distribution has become more negative, with only 24% positive sample.

With this observation from outcome distribution, it is expected that as the step-ahead format increases, the performance of the respective predictor will decrease due to shortage of training data.

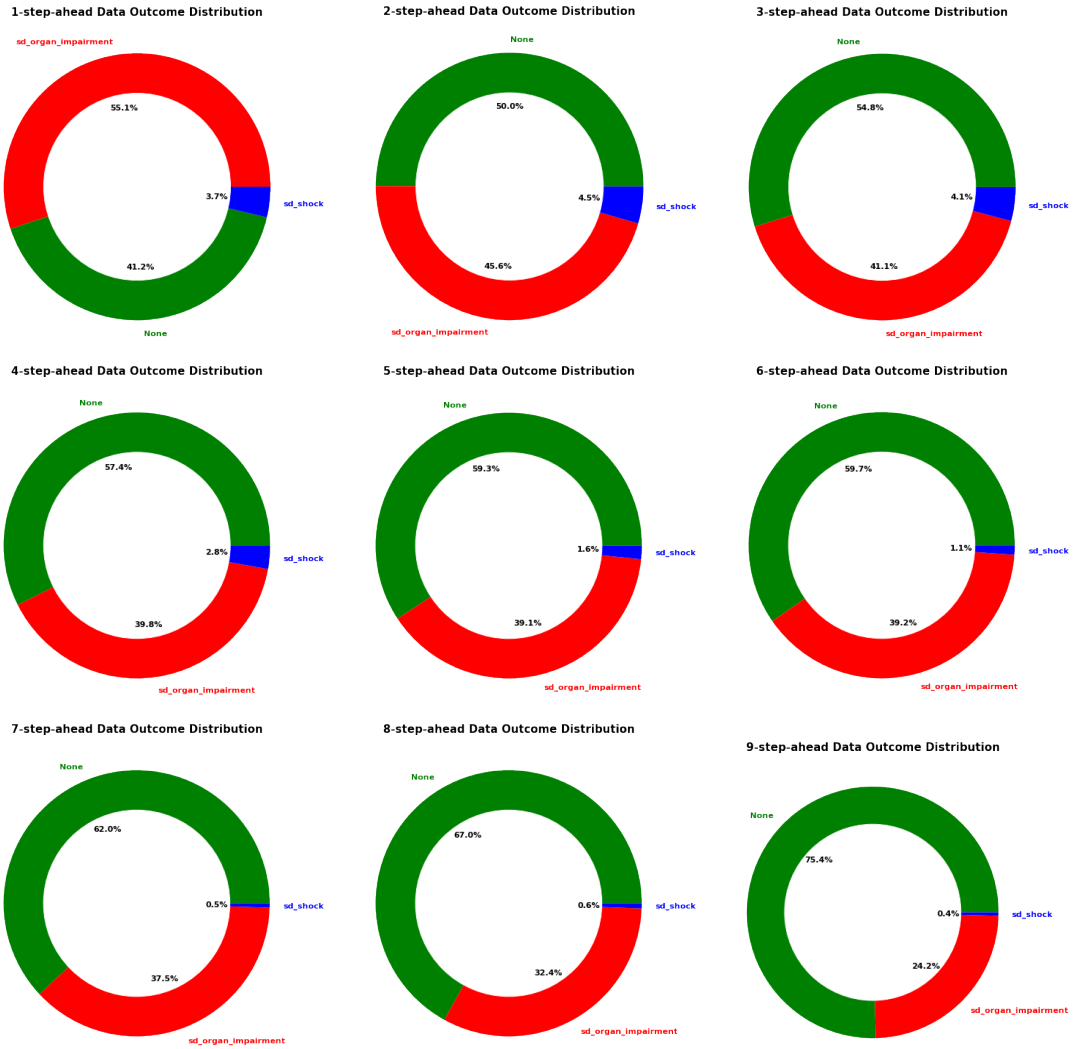


Figure 5.5: Outcome Distribution of Different Data Format

5.6.2 Missing Value Distribution

Figure 5.6 shows the distribution of missing/0 values in the data set. For this visualisation, strictly only numeric variables were used since binary variables contain 0 value that is meaningful. It can be observed that the amount of missing data is quite considerable.

Since the sequences of data are checked for all 0 values (which will be entirely dropped because it carries no useful information), the number of 0 values show in Figure 5.6 scatter between the nonzero values throughout the entire data set. Because of this, the

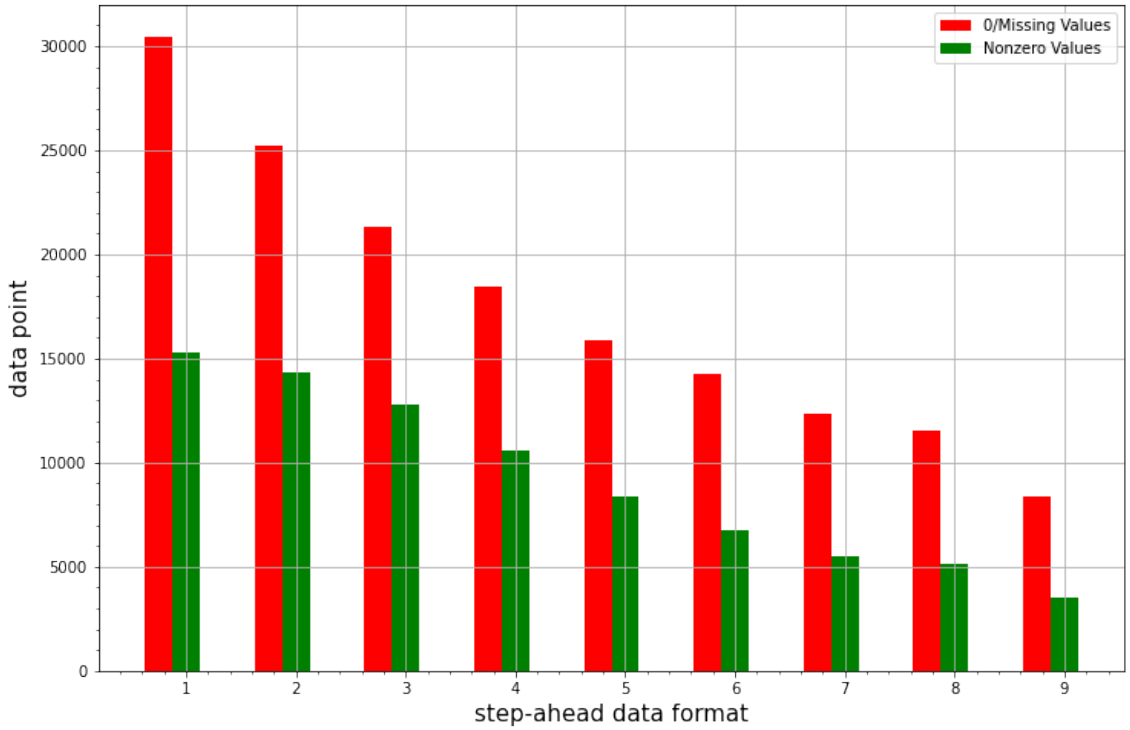


Figure 5.6: Distribution of Missing Values in Numeric Data

time series nature of data could be slightly disrupted. However, with the first predictors that have more than 15,000 non-zero data points to train on, there is ground to believe that the LSTM algorithms may be able to pick up the dependency between features and through times.

Chapter 6

Neural Network Architecture & Hyperparameter

6.1 Neural Network Architecture

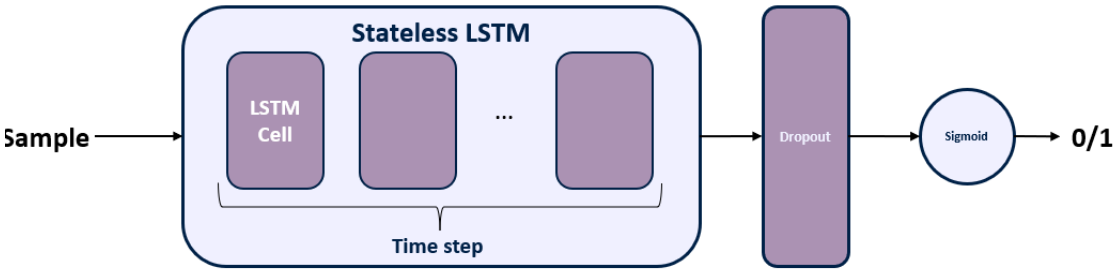


Figure 6.1: Neural Network Architecture

Figure 6.1 visualises the architecture of the overall neural network used to train the Severe Dengue Predictor. The network consists of a stateless LSTM followed by several Dense layers. The stateful LSTM is used when the entire data set is all related or dependent on each other in a timing manner. In that case, the final output of the previous batch will be carried to the next batch. In this project, there is no timing dependency between samples of data, or between patients. As a result, stateless LSTM is used.

A LSTM layer consists of multiple LSTM cells, which was introduced in Section 4.2. The

amount of LSTM cells depends on the `time_step` of the sample going through the layer. Each LSTM cell has a predefined number of units, `num_units`, which will determine the dimension of LSTM layer output. The unit of each LSTM cell can be thought as memory unit intuitively. A small `num_units` will not give LSTM enough memory space to store all the learnings from the sample. On the other hand, a larger `num_units` will be computationally expensive. To decide `num_units`, it is useful to look at the amount of data available for a starting point and do a few trials to pick out an optimal value, which is usually between $16 - 256$

6.2 Hyperparameter Tuning

6.2.1 Epoch

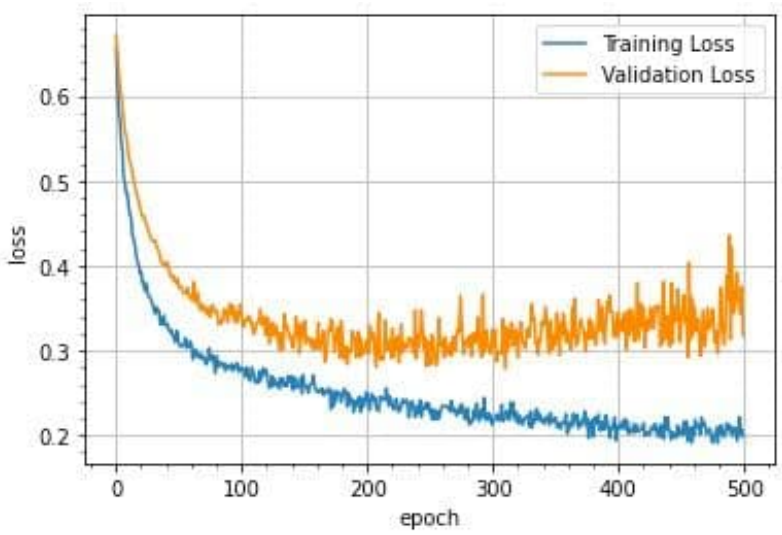


Figure 6.2: Training and Validating Progression

Epoch is the number of times the entire training data set is fed into the network. It is essential that the number of epoch is big enough for the network to learn and optimise its weights. However, it is also essential to reduce unnecessary computational expenses. As such, it is important to determine the appropriate number of epoch, which can be done through evaluating the progression of the model's loss with respect to epoch.

Figure 6.2 shows the loss of a sample training data set going through the proposed LSTM network. The validation loss reaches its minimum loss at 200 epochs, then starts to plateau until roughly 350 epochs, and finally starts to increases up until 500 epochs. This proves that with unnecessarily large number of epoch, not only it would be computationally expensive to train the model, the model is also more prone to overfitting. From the observation, 200 epochs will be used to train the models.

6.2.2 LSTM Unit

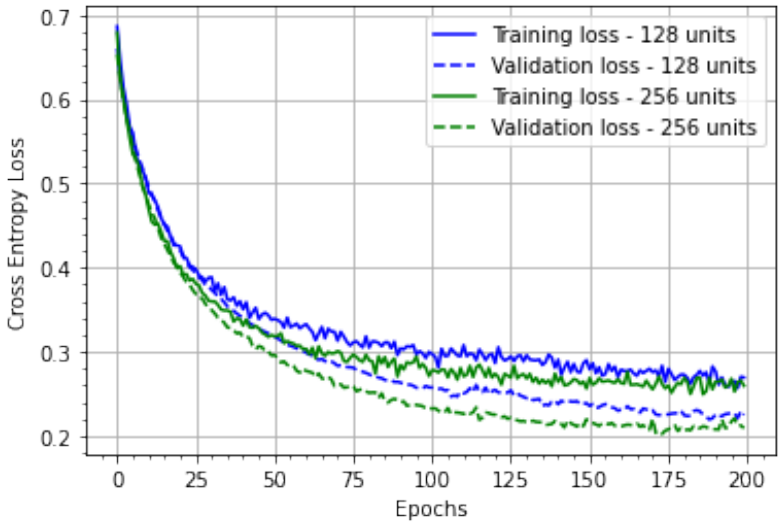


Figure 6.3: Performance Comparison of Models with Different LSTM Cell Units

Since the proposed neural network is fairly simple, with only 1 layer of LSTM and 1 layer of output Dense layer, the complexity should be focused inside of the LSTM. As mentioned previously, the usual value ranges between $16 - 256$. For architecture with parallel LSTM layers, it is not necessary to have a lot of units inside each LSTM layers because the number of LSTM layers will contribute greatly to the neural network's complexity. However, with a simple model proposed in Figure 6.1, we will look at using more units to help the learning ability of the model.

From Figure 6.3, we can see that the resulted losses between the two models are not very different. At the end of the 200 epochs, it is barely noticeable. Furthermore, it 256 units is more computationally expensive than 128, with 9 and 6 minutes fitting the model in Keras, respectively. As a result, 128 units will be considered for the LSTM layer.

6.2.3 Learning Rate

Learning rate η is also known as the step size moving towards the optimal point. Learning rate is important when optimizing the model and its loss. A too small step size can be computationally expensive to reach the optimal point. Meanwhile, a big step size may lead to the algorithm completely pass over and miss the optimal point.

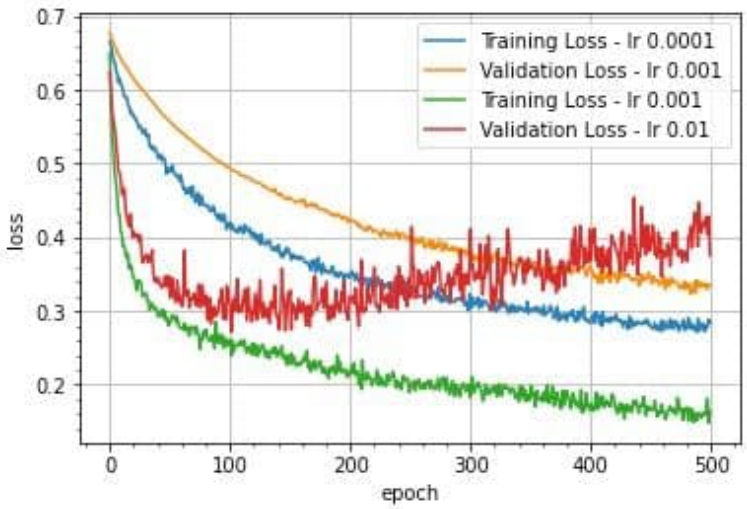


Figure 6.4: Training and Validating Progression for Different Learning Rate

Figure 6.5 shows the comparison between two different learning rates, 0.0001 (recommended by the founder of RMSProp optimizer) and 0.001. Since 0.0001 is quite a slow learning rate, it will be compared against a faster one to determine if it can help getting to the optimal point faster.

Unfortunately, the faster learning rate did not work out very great. Initially, validation loss of $\eta = 0.001$ converged much faster than that of $\eta = 0.0001$. However, after a few more

epochs, the algorithm completely missed the optimal point and overshot its loss without the ability to bounce back anytime soon. As a result, it is better to stay safe with the recommended learning rate to avoid overshooting.

6.2.4 Dropout

For dropout, values between 0.2-0.8 are tested to find out the optimal value. It is observed that for the largest dropout rate, 0.8, both training and validation losses are the highest. It means that for a dropout rate of 0.8, too many nodes were being disabled and the model cannot function properly. Dropout of rate 0.6 also has the lower performance. Between the dropout rate of 0.2 and 0.4, there is minimal difference. However, it seems that a dropout rate of 0.4 produced a model that's less overfitting than 0.6. As a result, dropout rate of 0.4 will be used for the architecture. This result shows that the model does not have a big overfitting problem, and only needs a minimal amount of regularisation to help improve itself.

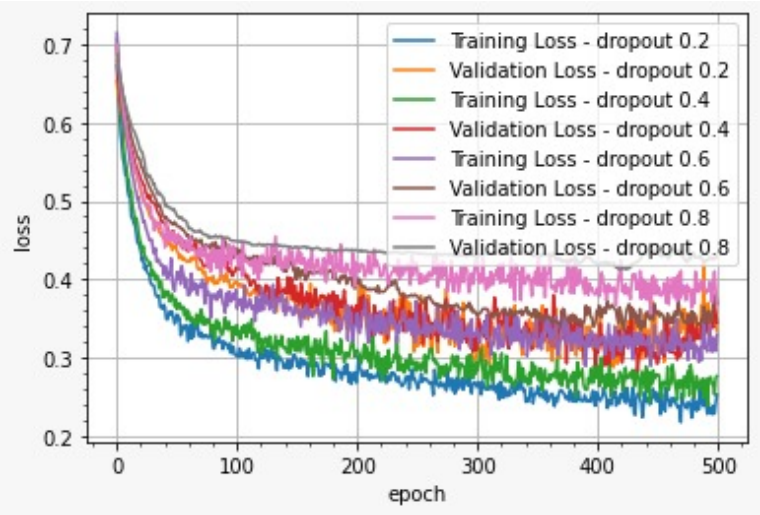


Figure 6.5: Training and Validating Progression for Different Dropout Rate

Chapter 7

Preliminary Results

7.1 Dynamic Severe Dengue Predictors

There are 9 predictors being developed and tested in this project, with increasing steps. The reason why it's called "step" instead of "day" is because when inspecting the data, the `day_from_onset` which indicates the time of record shows that not all records for patients are made daily. In reality, it is reasonable to expect that "step" and "day" are similar in this case.

	Accuracy	**Precision**	**Sensitivity**	**Specificity**	**PPV**	**NPV**
1-step-ahead	95 ± 2	96 ± 2	96 ± 4	94 ± 3	95 ± 3	95 ± 6
2-step-ahead	95 ± 1	93 ± 2	98 ± 2	92 ± 3	91 ± 2	98 ± 2
3-step-ahead	89 ± 3	86 ± 6	92 ± 4	90 ± 4	85 ± 6	92 ± 3
4-step-ahead	86 ± 3	84 ± 3	84 ± 5	87 ± 3	81 ± 4	86 ± 4
5-step-ahead	86 ± 2	82 ± 4	85 ± 5	87 ± 5	80 ± 5	88 ± 3
6-step-ahead	85 ± 2	81 ± 4	82 ± 4	86 ± 3	78 ± 6	86 ± 3
7-step-ahead	79 ± 4	72 ± 6	73 ± 12	82 ± 4	70 ± 5	82 ± 6
8-step-ahead	83 ± 2	75 ± 3	75 ± 9	82 ± 5	65 ± 8	84 ± 9
9-step-ahead	85 ± 2	83 ± 5	53 ± 12	89 ± 1	77 ± 9	86 ± 3

Table 7.1: Performance of 9 Dynamic Severe Dengue Predictor using Maximum Data Length

Initially, the predictors are formatted to use the maximum data length amongst the entire data set as the common data length. This means that none of the patients will get

deleted which will lead to data loss. Since the outcome distribution of the data is lacking which is thought to be ideal to use the format that would preserve the largest amount of data possible.

Table 7.1 shows that the longer step-ahead predictors have steadily decreasing performance, especially sensitivity metric, compared to the previous ones. The reason for the decreasing performance can be explained by the data set size. As the step-ahead increases, the number of rows of data getting trimmed increases. And eventually there would be patients that do not have enough rows of data to contribute to the predictor. Unfortunately, a lot of the data getting cut out happen to be positive samples. The entire data set started out on the more positive side, where at 1-step-ahead format, the number of positive samples is 43% more than the negative samples. However, as the data gets dropped, the data become more on the negative side, with the number of negative samples becomes 3 times as many as the positive samples at 9-step-ahead format. This outcome distribution explains why the sensitivity metric decreases its performance faster than any other metric. Table 7.2 also shows the amount of data decreasing between each predictor. From 1 to 9-step-ahead predictor, more than 65% percent of the samples have been dropped because of inadequate rows of data.

Step-ahead Predictor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Total Data Set Size	1330	1097	998	948	911	875	765	642	455
Positive Data Size	782	549	451	404	370	353	291	212	112
Negative Data Size	548	548	547	544	540	522	474	430	343

Table 7.2: Data Set Size Used to Train Each Predictor

Overall, all predictors have adequately high specificity and slightly lower sensitivity. This is to be expected because there is always a trade-off between sensitivity and specificity. This problem is also partially because of the distribution of positive/negative sample distribution as shown in Table 7.2. As the step-ahead increases, the amount of data decreases. However, overall, the amount of negative sample does not change as drastically as with positive sample, which explains why the predictors' ability

to correctly identify a true negative sample is adequately high (demonstrates by specificity metric). Meanwhile, since the positive sample is seriously lacking as the step-ahead increases, the predictor's ability to correctly identify true positive sample depreciates quickly, to only equal to a random guess at 9-step-ahead, with 53% sensitivity.

PPV and NPV performance of predictors are also on the adequately high side. One of the aims when developing Severe Dengue Predictors is clinical management which helps prioritise the more vulnerable patients and relieve burdens on the hospital system during a mass outbreak. The positive predictive value (PPV) shows the confidence that if a patient is tested positive with Severe Dengue Predictor, there is up to 92% he/she will develop Severe Dengue. This ensures a high probability that patients are not wasting time in hospitalisation when getting waiting for symptoms, as well as hospitals not wasting too many resources on hospitalising patients with mild progression. Furthermore, with comparatively high negative predictive value (NPV), up to 98% of patients getting negative prediction stays safe and does not progress to Severe Dengue.

With the overall performance of all predictors, it is sufficient to say that the warning signs and some highly correlated features carry enough information to adequately train the predictors with simple architecture.

7.2 Sensitivity Analysis

Sensitivity analysis is a technique used to determine if the data set is sufficient to estimate the performance and generalisation power of a model and devise a future plan for the project to further improve.

Figure 7.1 shows that the sensitivity analysis result for each predictor varies differently, which is expected since the numbers of data available for each predictor also vary greatly. From observation, 1-step-ahead predictor variance is very low, which means that the

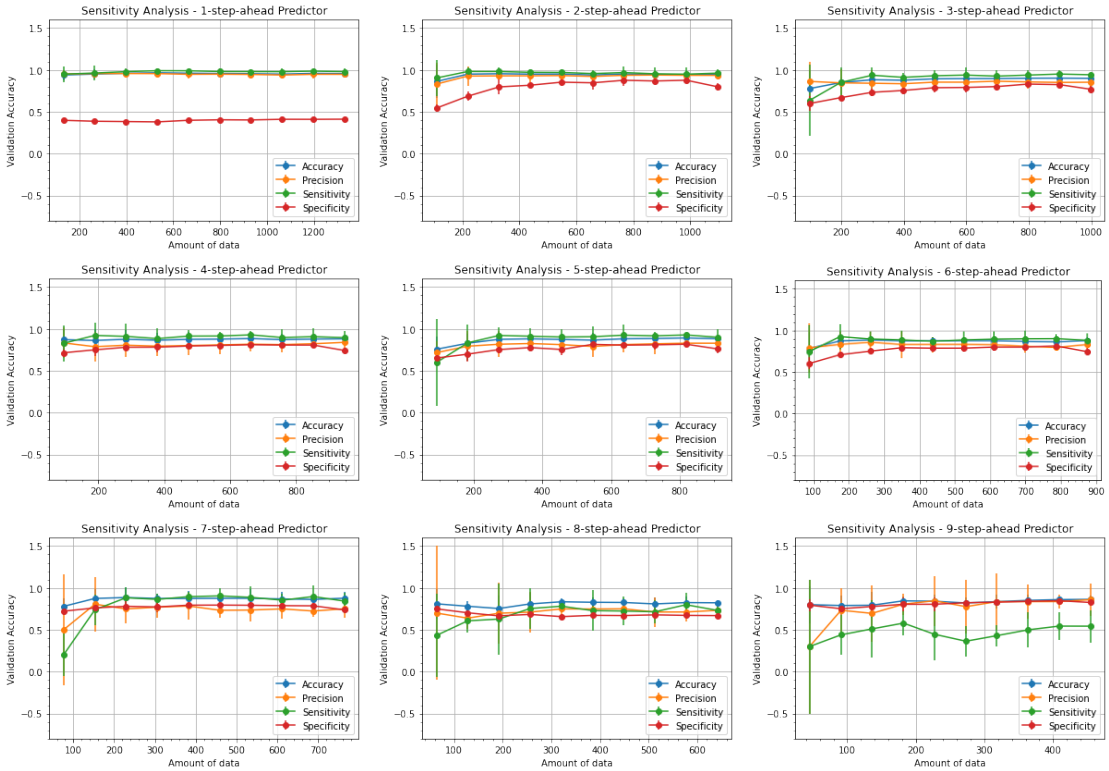


Figure 7.1: Effect of Data Set Size on Predictors' Performance and Variance

amount of training data is sufficient to estimate the performance of this predictor in the real world. The same can be said for 2 and 3-step-ahead predictors. However, for the remaining ones, such as 4, 5, 6, 7, and 8-step-ahead predictors, there are still noticeable variances in the performance which indicates that the model will benefit greatly from having a larger data set. Meanwhile, the 9-step-ahead predictor desperately needs more data, which was also heavily emphasised in Section 5.6.1.

7.3 Improving Preliminary Model

Since the further step-ahead models were performing badly because of imbalanced data, a way that may improve the performance is to assign different weights to the labels. Assigning weight will affect the loss equation as follow:

$$class\_weight(\hat{y}^{(t)}, y^{(t)}) = \begin{cases} \mathbb{I}(\hat{y}^{(t)} \neq y^{(t)}) = \alpha & \text{if } y^{(t)} = 1 \\ \mathbb{I}(\hat{y^{(t)}} \neq y^{(t)}) = 1 & \text{if } y^{(t)} = 0 \end{cases} \tag{7.1}$$

$$\mathcal{L}(\hat{y}, y) = \sum_{1}^{T} \Big[class\_weight(\hat{y}^{(t)}, y^{(t)}) * BCE(\hat{y}^{(t)}, y^{(t)}) \Big]$$

Where α is the weight assigned to positive class, and the weight for negative class remains at 1. Since the positive sample is lacking in numbers compared to the negative sample, assigning $\alpha > 1$ will essentially tell the loss function and optimizer that a positive sample is α times more important than a negative sample. When a positive sample is classified wrong, the predictor will be "punished" more than when a negative sample is classified wrong.

	Accuracy	**Precision**	**Sensitivity**	**Specificity**	**PPV**	**NPV**
3-step-ahead	90 ± 2	85 ± 5	96 ± 2	89 ± 4	84 ± 6	95 ± 2
4-step-ahead	88 ± 3	85 ± 4	87 ± 3	87 ± 4	82 ± 4	88 ± 4
5-step-ahead	87 ± 3	81 ± 5	91 ± 6	85 ± 6	78 ± 7	91 ± 4
6-step-ahead	85 ± 1	79 ± 5	87 ± 6	85 ± 5	77 ± 6	89 ± 3
7-step-ahead	80 ± 3	70 ± 4	84 ± 6	79 ± 3	67 ± 4	87 ± 5
8-step-ahead	81 ± 2	69 ± 5	82 ± 7	83 ± 3	62 ± 11	90 ± 3
9-step-ahead	86 ± 5	73 ± 14	75 ± 10	86 ± 5	67 ± 15	92 ± 3

Table 7.3: Performance of Predictors when Training with $\alpha = 1.5$

Table 7.3 shows the performance of predictors with imbalanced training data when trained with weight $\alpha = 1.5$ for positive class. When comparing with Table 7.1 where the predictors have been trained without weight, the sensitivity metric has improved a noticeable amount, especially with the 9-step-ahead model. However, the standard deviation between each iteration in K-fold cross-validation is also quite large. In Table 7.3, the specificity and PPV metric has been decreased quite a significant amount, especially in PPV. This means that the number of false positive prediction has gone up considerably. However, the performance of NPV has increased. With the new performance, the predictors still have power to aid clinical management in ensuring that patients tested negative via Severe Dengue Predictor will have at least 87% chance of not developing

Severe Dengue, thus allowing them to feel safe when getting treatment at home and not taking up or spending on unnecessary hospitalisation. However, with quite low PPV, the rate of unnecessary hospitalisation after the 5-step-ahead prediction is still considerable.

Chapter 8

Model Deployment

To assist clinical trial of the machine learning models, all predictors were deployed using Flask web service. Since it is a very first deployment, and the nature of hospital's available database service is unavailable, the application is designed to take clinical inputs from clinicians and return predictive results over the next 9 days.

Figure 8.1: Model Deployment UI

8.1 Application Development

Tensorflow and Keras API allow downloading the trained neural network model in the `h5` format. When downloading, the file contains constructed architecture, weight values, and `compile()` information such as optimizer, learning rate, batch size, number of epochs, and other evaluation metrics.

For this project, the processing pipeline consists of formatting data and standardisation using `RobustScaler`. The scaler is previously fitted with training data and when downloaded as `pkl` format, it contains interquartile ranges and mean value of the training data. The input data provided by the user will be standardised based on this information.

With the processing pipeline and trained neural network available, a straightforward algorithm can be implemented for the application as shown in Figure 8.4.

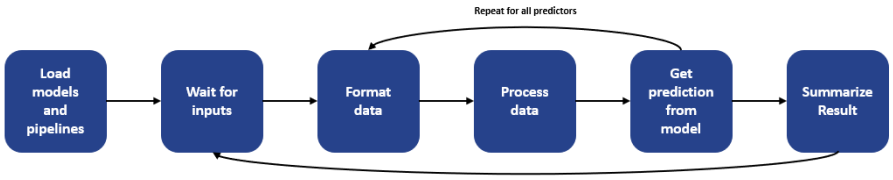


Figure 8.2: Model Deployment Algorithm

8.2 Application Design

The design of application is simple and straight to the point, with forms to fill out clinical data of 1 patient. Each row represents 1 day worth of clinical data. If the data is not available for that day i.e., some tests were not carried out that day, then the value can be 0. For multiple-day data, an `Add Row` button can be used to expand the form and enter multiple-day data. Figure 8.3 shows an example of data being input into the application. These 2 days' worth of data were taken from patient 1888 who has eventually developed severe organ impairment after 10 days worth of record. The full data of patient 1888 can

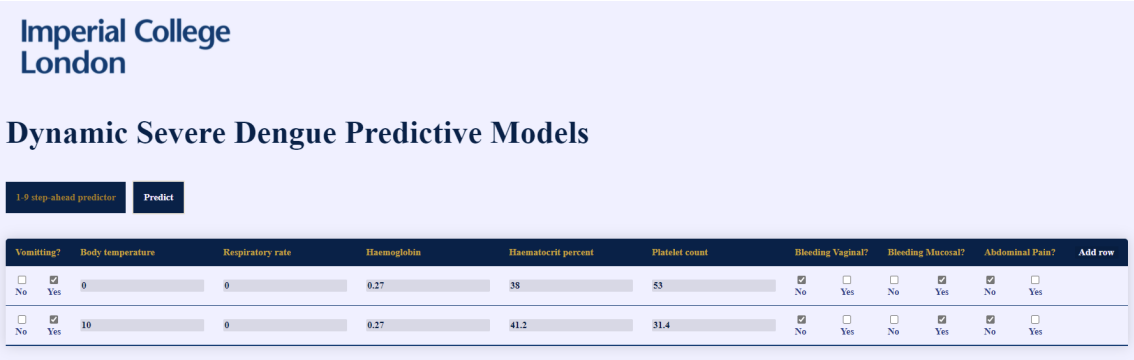


Figure 8.3: Application Input Page Design

be found in Appendix B.

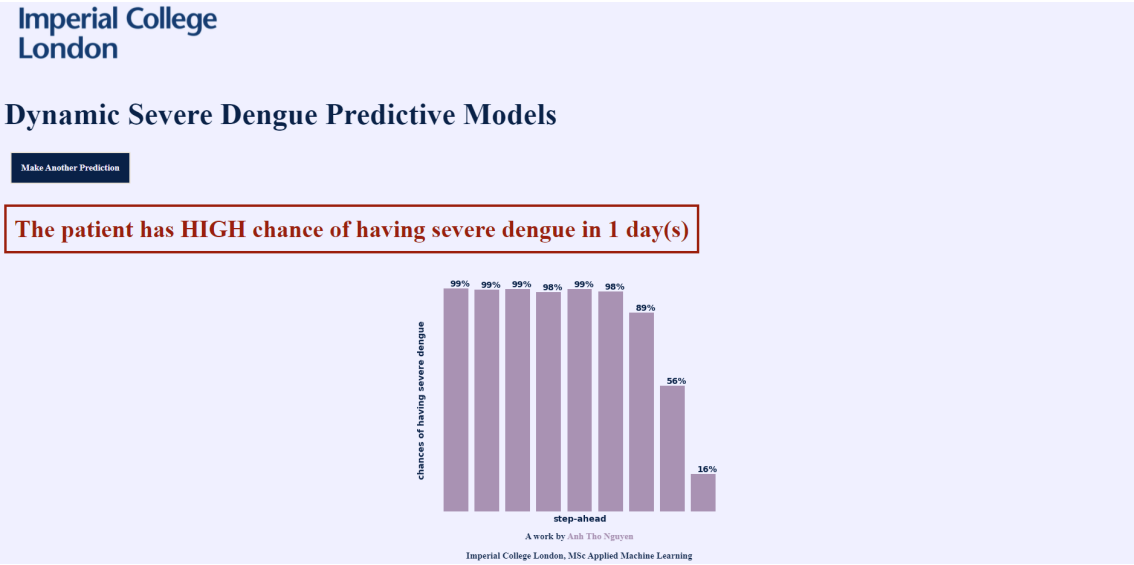
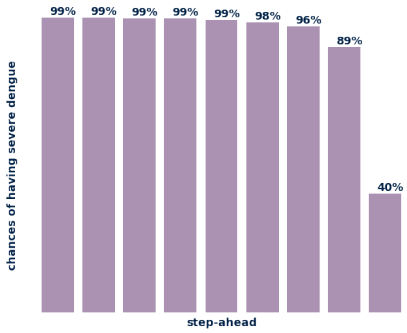


Figure 8.4: Application Output Page Design

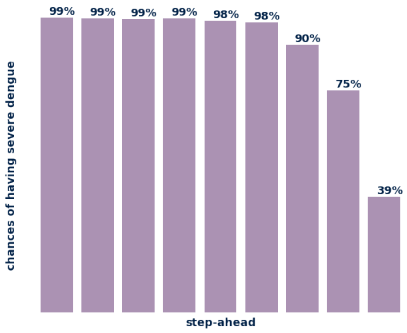
After the patient's data has been filled correctly, by clicking the `Predict` button, the algorithm will start collecting inputs and entering the loop to obtain prediction results over the next 9 steps (days) ahead. The result page consists of 2 important parts: the warning text and the chart showing the chances of developing Severe Dengue over the next few steps/days. The warning text works based on a decision threshold of 50%. Thus, any result greater than 50% will be marked as having a high chance of developing Severe Dengue. By providing a chart with more specific result from predictor, it will help clinicians who have gained experience from using the predictors to enhance their diagnosis.

8.3 Application Simulation

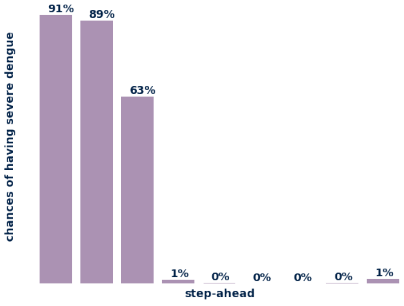
8.3.1 Positive Sample Simulation



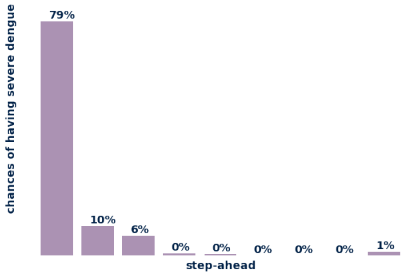
(a) Patient 1888 - 1 Day Data



(b) Patient 1888 - 2 Days Data



(c) Patient 602 - 1 Days Data



(d) Patient 602 - 2 Days Data

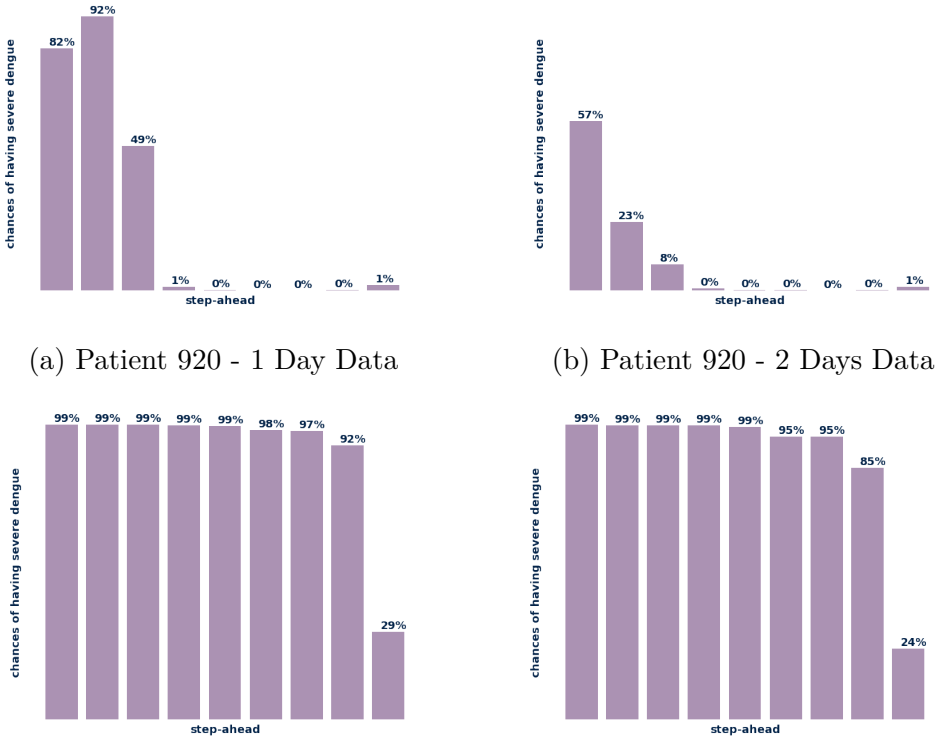
Figure 8.5: Simulation on Positive Samples

Figure 8.5 shows the results when simulating up to 2 days' worth of patients' data to the application. The 2 samples used in this simulation is from patient 1888, who developed Severe Dengue after 6-7 steps (days) subsequently. From the simulation of 1 day's worth of data, it seems very promising since the predictors gives very high probability of developing Severe Dengue up to 7-step-ahead predictor. This signals that the predictors are very sensitive to positive samples. In Figure 8.5b, the application again shows very promising results when after the 6th predictor, the probability of patient developing Dengue starts to decrease quite noticeably.

Another sample shown in this simulation is of patient 602, who subsequently developed Severe Dengue 1-2 days after these data were recorded. For only one data as shown in Figure 8.5c, the predictors were able to narrow down roughly the date that the patient would develop Severe Dengue (2 days after, in this case). Similarly, Figure 8.5d correctly shows that patient 602 would develop Severe Dengue the day after.

From these simulations, we observed a highly accurate and precise application that were able to identify if the patient will develop Severe Dengue, and roughly narrow down the day that Severe Dengue will happen. As more data is fed into the predictors, the accuracy increases.

8.3.2 Negative Sample Simulation



(a) Patient 920 - 1 Day Data

(b) Patient 920 - 2 Days Data

(c) Patient 934 - 1 Days Data

(d) Patient 934 - 2 Days Data

Figure 8.6: Simulation on Negative Samples

Figure 8.6 shows the result of 2 steps' (days) worth of data from 2 different patients who did not develop Severe Dengue. Patient 920 results shows that as more data are fed into the models, it accuracy becomes better. With this trajectory, in just 3 steps (days) since onset, the patients can be confident that they would not develop Severe Dengue.

However, for patient 934 shown in Figure 8.6c & d, the predictors were getting very confused. They were adamant that the patient would definitely go on to develop Severe Dengue when in fact, he or she won't.

This simulation shows that for negative samples, some of the data will not work. This is also shown in the performance table of predictors in Figure 7.1 where it shows the PPV metric is lacking compared to that of NPV metric. Meaning that the chances of patients getting a false positive is greater than getting a false negative.

Chapter 9

Updated Results

9.1 Alternative Data Formatting

	Accuracy	Precision	Sensitivity	Specificity	PPV	NPV
1-step-ahead	95 ± 1	97 ± 1	97 ± 1	79 ± 10	97 ± 2	80 ± 5
2-step-ahead	94 ± 1	95 ± 1	97 ± 2	85 ± 4	95 ± 1	94 ± 4
3-step-ahead	93 ± 3	93 ± 4	97 ± 2	84 ± 8	92 ± 4	93 ± 4
4-step-ahead	92 ± 3	91 ± 2	97 ± 2	85 ± 7	90 ± 2	93 ± 5
5-step-ahead	91 ± 2	88 ± 3	99 ± 1	83 ± 6	86 ± 3	98 ± 2
6-step-ahead	89 ± 2	83 ± 3	97 ± 2	82 ± 4	85 ± 3	96 ± 3
7-step-ahead	85 ± 4	81 ± 3	88 ± 6	78 ± 4	81 ± 4	88 ± 5
8-step-ahead	83 ± 5	70 ± 9	67 ± 13	87 ± 1	66 ± 16	88 ± 4
9-step-ahead	81 ± 3	36 ± 37	15 ± 15	88 ± 3	26 ± 22	83 ± 3

Table 9.1: Performance of Predictors when Training with Median Data Length

From the simulation in the previous chapter, it is thought that because the maximum-data-length format is used when training models, the models were having problems in simulation when receiving little amount of data than normal, which is more common in real life than receiving 10 steps' (days) worth of data. It is unreasonable because because on average, Severe Dengue occurs 3-7 days after illness onset, which is why in the simulation, only up to 2 steps (days) worth of data is fed into the application. With that observation, the common data length of training data is changed to median instead of maximum data length. This will create truer scenarios to real life (though not perfect) where only a very limited number of data is input into the application for efficiency.

Table 9.1 shows the performance of models trained with median-data-length format. Compared to the original model in Table 7.1, mostly all the metrics indicate that the performance of predictors have increased, even when the amount of training data has decreased, as shown in Table 9.2. The progress of data distribution is similar to that of maximum-data-length format. However, in Table 9.1 at 1-step-ahead format, median data length has a more skewed positive distribution than that of maximum-data-length, which explains why the specificity and its standard deviation were low initially.

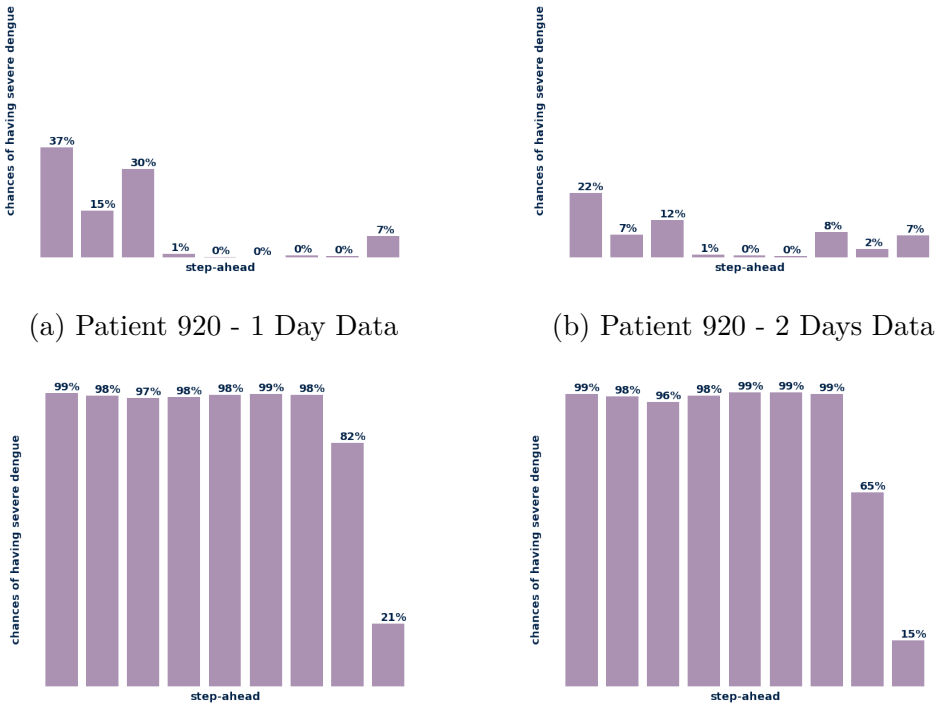
Step-ahead Predictor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Total Data Set Size	822	589	623	573	536	546	394	369	237
Positive Data Size	663	430	398	351	318	268	190	102	42
Negative Data Size	159	159	225	222	218	278	204	267	195

Table 9.2: Data Set Size Used to Train Each Predictor with Median-Data-Length Format

Overall, the label distribution of median-data-length format is slightly less skewed than that of maximum-data-length. However, from 8-step-ahead format, the data was seriously lacking since some of the data has initially been dropped when formatting median-data-length, then a considerable amount has also been deleted when formatting data as 8-step-ahead. The skewed distribution and lack of data explain why the performance of these two predictors (8 and 9) is especially worse than that of maximum-data-length format.

9.2 Further Simulation

With the updated model, the simulation in Section 8.3.2 is repeated to see if there is any performance progress in simulation. Patient 920 who has previously been adequately identified as negative, has been given better result this time using predictors trained on median-data-length data format. From only data of the first day, the predictors have all correctly identified that patient 920 would not develop Severe Dengue.



(a) Patient 920 - 1 Day Data

(b) Patient 920 - 2 Days Data

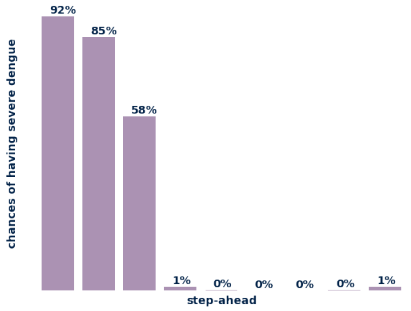
(c) Patient 934 - 1 Days Data

(d) Patient 934 - 2 Days Data

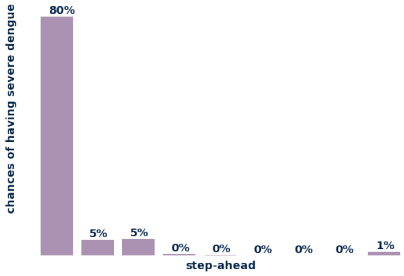
Figure 9.1: Further Simulation on Previous Negative Samples using Median Data Length

Unfortunately, the simulation of patient 934 did not progress any better. However, when continue simulating other negative sample, a considerable increase in performance can be observed in some cases where the data is not "extreme". Figure 9.2 shows that for the same amount of data input to the applications, the median-data-length trained predictors outperformed the previous predictors. With only 2 days worth of data, which consists of almost 30% missing values, the later predictors were able to produce highly accurate predictions.

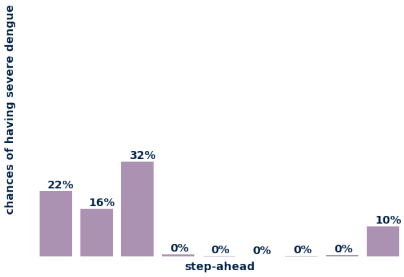
From the updated simulation result of patient 920, 934, and 641, it can be safely said that the changing from maximum data length to medium data length has increased the performance of predictors, even in simulation. However, there are blind spots which will trigger the predictors to associate those spots with positive label most of the times. This



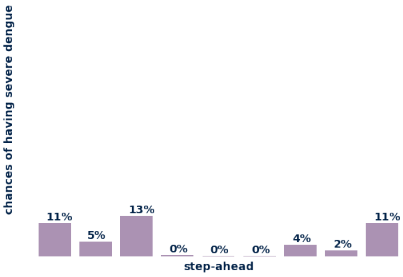
(a) Maximum Data Length - 1 Day Data



(b) Maximum Data Length- 2 Days Data



(c) Median Data Length - 1 Days Data

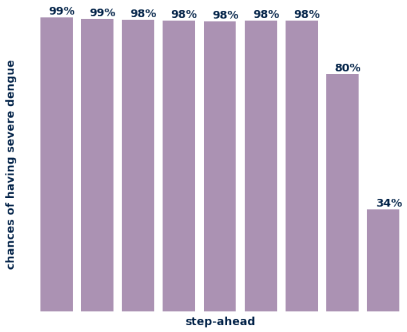


(d) Median Data Length - 2 Days Data

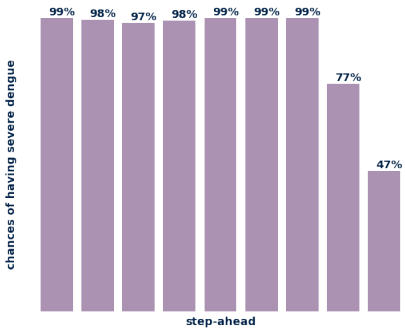
Figure 9.2: Performance Difference between Median and Maximum Data Length on Negative Sample of Patient 641

problem could have been arisen because there are not enough data to expand LSTM's hypotheses.

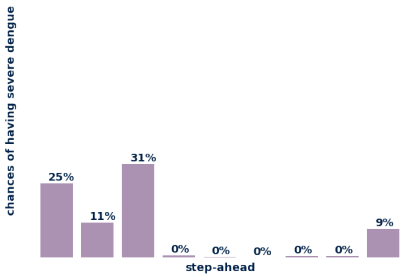
Figure 9.3 shows the results of simulating positive samples on models trained with median-data-length format. Unfortunately, for these 2 samples, there were not any noticeable increase in performance. Predictive result for patient 1888 did not change drastically. On the other hand, results for patient 602 are both wrong this time, even though it was correctly predicted when using models trained with maximum-data-length format. The reduction in performance is consistent with the decrease in NPV metric of 1 and 2-step-ahead predictors when trained on median-data-length.



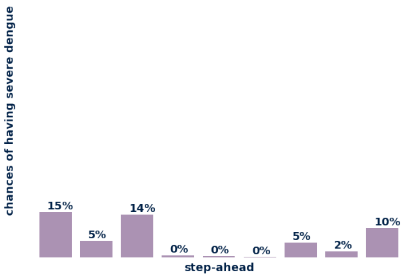
(a) Patient 1888 - 1 Day Data



(b) Patient 1888 - 2 Days Data



(c) Patient 602 - 1 Days Data



(d) Patient 602 - 2 Days Data

Figure 9.3: Further Simulation on Positive Samples using Median Data Length

Behind the scene, multiple random simulations have been performed. The results of those simulations suggest that the NPV rate may be higher than PPV in real life. And even though the simulation of patient 602 in Figure 8.5 suggests that the machine learning application may be able to narrow down the rough date that Severe Dengue would develop, other simulations suggest that it is only a rare occurrence. The majority of correctly predicted results on positive samples look like Figure 9.3a and b where all almost all predictors show high chance of patient developing Severe Dengue.

Chapter 10

Discussion

The performance of LSTM-based architecture on different types of data proves that the concept of using clinical data on a recurrent neural network based architecture is feasible. The network was able to perform fairly well on adequate training data size, with remarkably high sensitivity, specificity, PPV and NPV metrics. The performance is quite surprising since the training data set has at most 50% missing values without any data interpolation. LSTM has been able to identify the dependencies between variables, through times, and create different hypotheses for different dependencies.

10.1 Data Set

Since Severe Dengue can occur 3-7 days after illness onset, it is important that we have at least a 7-step-ahead predictor working with adequate performance to aid the clinical management during outbreak. However, one of the biggest obstacles is the lack of data, either through each data point (missing values) or through each patient/participant. Furthermore, data imbalance is also a huge problem proven when looking at the predictors' performance. It does not help since predicting further step-ahead will undoubtedly be more complicated than 1-step-ahead. As a result, having more data points, and more patients that can present different trajectories for the development of Severe Dengue

through several days will be very important. Now that a small clinical feature set that has great impact on predicting Severe Dengue has been identified, it may reduce the amount of tests clinicians would have to collect for the data set study.

A method that has been trialed in this project to potentially solve the problem of imbalanced data set is assigning weights to labels. While the sensitivity of predictors increases, the PPV metric unfortunately decreases. This means that there will be a trade-off when using this method. For the purpose of clinical management, a high PPV and NPV will more preferable compared to sensitivity and specificity.

Another limitation of the data set used in this project aside from lack of data, is that it includes only Vietnamese children between ages 5-15. This will undoubtedly hinder the generalisation power of predictors as other race and age range can experience Dengue progression on different levels.

10.2 Performance Comparison

Compared to DARE & DART calculators covered in Section 3.2, Severe Dengue Predictors has greatly improved the specificity and PPV metrics, while still keeping the sensitivity and NPV metrics at an adequate level. This is very important, especially when taking into account the use of Severe Dengue Predictors in healthcare settings. The main aim when developing the predictors is to not only help doctors in diagnosis, but also helping patients relieve the panic that they may potentially develop Severe Dengue and the best thing for them to do is to get hospitalised immediately.

Compared to the Early Severe Dengue Identifier (ESDI) covered in Section 3.3, Severe Dengue Predictors are trained on a more complex model, using more relevant features chosen according to WHO Dengue guidelines. Because of that, the performance of Severe

Dengue Predictors is slightly better. However, it is hard to quantify the difference in performance of both models because ESDI predicts the chances Severe Dengue will develop in an infinite future, while Severe Dengue Predictor predicts certain steps (days) ahead.

Through random simulations, it is verified that the application also has great potential to be used in real life to help diagnosing Severe Dengue, even when using only a few days' worth of data (1-3 days) in the application. In rare cases, the application shows the ability to correctly narrow down the date that patient may develop Severe Dengue. However, most of the correctly predicted result for positive samples usually shows high chances of developing Severe Dengue from day 1-7.

After performing simulation, it is observed that in multiple instances, the application has completely diverged from the true label. When re-training models using median-data-length format, the problem was relieved in some instances. However, in some extreme examples, it did not get any better. This led to a possible assumption that the data set failed to create enough scenarios for the models to build and expand their hypotheses on. Thus, when presented with the extreme examples, the predictors use their incompletely trained hypothesis to produce results which may turn out to be very far from the ground truth. Through those simulations, it is suggested that the application has greater NPV rate than that of PPV.

10.3 Future Work

For future work, a model trained on imputed data set would be a good starting point. Throughout this project, models were trained with missing values and still show promising result. It is anticipated that when using data imputed with proper interpolation technique, the results will increase significantly, especially on further step-ahead predictors

such as 8 and 9-step-ahead, since currently they are having a serious problem of lacking data.

Other data formats which promote real-life scenario such that user usually only inputs a few days' worth of data into the model instead of 10 days, should also be explored. While the performance of predictors seem good from K-fold cross-validation, it is when doing simulation that some limitations of application in real-life scenario were noticed.

Furthermore, instead of binary classification, a regression model that predicts the day Severe Dengue is worth looking into. In this project, the regression model was attempted. However, date features were either missing a lot of data points or incomprehensibly recorded, which hinder the feasibility of developing and training the regression model.

An improvement that would make the use of this machine learning model deployment more sufficient is to connect the application with any database that the hospital uses to store patient's clinical data. With this design, clinicians will not have to manually enter each patient's data into the application. The clinical data will be loaded automatically, data processing and prediction will be made behind the scene and constantly get updated as more relevant information come in. When the clinician needs to make a diagnosis, he or she can simply pull up the patient's record and a note from the predictor attaching predictive values will assist him or her in making the decision.

Chapter 11

Conclusion

Dengue is a mosquito-borne viral infectious disease that has severe impacts on not only the patients but also on the entire endemic country's economy. With global warming and other side effects of technology, Dengue has spread throughout the world and began wreaking havoc on parts of the world that have never experienced Dengue outbreak before. Understanding how serious a Dengue outbreak can be without proper preparation, this project aims to develop a machine learning model that would contribute to the battle against Dengue reactively.

Using knowledge from previously developed Dengue machine learning models, multiple version of Severe Dengue Predictors were developed, trialed, and tested. The best predictor can reach a 96% sensitivity, 94% specificity, 95% PPV and NPV, and the mean performance of 1 to 9-step-ahead predictors is 82%, 88%, 80%, and 89%, respectively, for models trained with maximum-data-length.

The working models were then deployed on a web-based application. Multiple trials which simulates the real-life scenario where users usually only input a few days worth of data are run, and real-life performance was partially verified to be adequate. Although, through multiple simulation, it is observed that NPV metric in application is greater than that of PPV metric.

This machine learning model is a step closer to control outbreak in a reactive way, efficiently. Combining with other methods to proactively control Dengue, doctor's experience and expertise, Dengue outbreak will have less effect on the economy and at-risk patients will receive proper treatment on time.

Bibliography

[1] V. N. News, "Dengue fever at risk of spreading across the country," 2016. [Online]. Available: https://vietnamnews.vn/society/301054/dengue-fever-at-risk-of-spreading-across-the-country.html

[2] J. Xia, S. Pan, M. Zhu, G. Cai, M. Yan, Q. Su, J. Yan, and G. Ning, "A long short-term memory ensemble approach for improving the outcome prediction in intensive care unit," *Computational and Mathematical Methods in Medicine*, vol. 2019, pp. 1–10, 2019.

[3] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[4] K. W. Tan, B. Tan, T. L. Thein, Y.-S. Leo, D. C. Lye, B. L. Dickens, J. G. X. Wong, and A. R. Cook, "Dynamic dengue haemorrhagic fever calculators as clinical decision support tools in adult dengue," *Transactions of The Royal Society of Tropical Medicine and Hygiene*, 2019.

[5] 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue

[6] M. Campos, D. Ward, R. F. Morales, A. R. Gomes, K. Silva, N. Sepúlveda, L. F. Gomez, T. G. Clark, and S. Campino, "Surveillance of aedes aegypti populations in the city of praia, cape verde: Zika virus infection, insecticide resistance and genetic diversity," *Parasites Vectors*, vol. 13, no. 1, 2020.

[7] C. J. Burrell, C. R. Howard, and F. A. Murphy, "Flaviviruses," *Fenner and White's Medical Virology*, pp. 493–518, 2017.

[8] F. J. Colón-González, L. Soares Bastos, B. Hofmann, A. Hopkin, Q. Harpham, T. Crocker, R. Amato, I. Ferrario, F. Moschini, and S. e. a. James, "Probabilistic seasonal dengue forecasting in vietnam: A modelling study using superensembles," *PLOS Medicine*, vol. 18, no. 3, p. e1003542, 2021.

[9] T. M. Hung, H. E. Clapham, A. A. Bettis, H. Q. Cuong, G. E. Thwaites, B. A. Wills, M. F. Boni, and H. C. Turner, "The estimates of the health and economic burden of dengue in vietnam," *Trends in Parasitology*, vol. 34, no. 10, pp. 904–918, 2018.

[10] 2021. [Online]. Available: https://healthinformatics.uic.edu/blog/machine-learning-in-healthcare/

[11] N. A. M. Salim, Y. B. Wah, C. Reeves, M. Smith, W. F. W. Yaacob, R. N. Mudin, R. Dapari, N. N. F. F. Sapri, and U. Haque, "Prediction of dengue outbreak in selangor malaysia using machine learning techniques," *Scientific Reports*, vol. 11, no. 1, 2021.

[12] P. K. Lam, T. V. Ngoc, T. T. Thu Thuy, N. T. Hong Van, T. T. Nhu Thuy, D. T. Hoai Tam, N. M. Dung, N. T. Hanh Tien, N. T. Thanh Kieu, and C. e. a. Simmons, "The value of daily platelet counts for predicting dengue shock syndrome: Results from a prospective observational study of 2301 vietnamese children with dengue," *PLOS Neglected Tropical Diseases*, vol. 11, no. 4, p. e0005498, 2017.

[13] P. K. Lam, D. T. H. Tam, T. V. Diet, C. T. Tam, N. T. H. Tien, N. T. T. Kieu, C. Simmons, J. Farrar, N. T. N. Nga, and P. T. e. a. Qui, "Clinical characteristics of dengue shock syndrome in vietnamese children: A 10-year prospective study in a single hospital," *Clinical Infectious Diseases*, vol. 57, no. 11, pp. 1577–1586, 2013.

[14] T. Dinh The, T. Le Thi Thu, D. Nguyen Minh, N. Tran Van, H. Tran Tinh, C. Nguyen Van Vinh, M. Wolbers, T. Dong Thi Hoai, J. Farrar, and C. e. a. Simmons, "Clinical features of dengue in a large vietnamese cohort: Intrinsically lower platelet counts and

greater risk for bleeding in adults than children," *PLoS Neglected Tropical Diseases*, vol. 6, no. 6, p. e1679, 2012.

[15] N. M. Tuan, H. T. Nhan, N. V. V. Chau, N. T. Hung, H. M. Tuan, T. V. Tram, N. L. D. Ha, P. Loi, H. K. Quang, and D. T. H. e. a. Kien, "Sensitivity and specificity of a novel classifier for the early diagnosis of dengue," *PLOS Neglected Tropical Diseases*, vol. 9, no. 4, p. e0003638, 2015.

[16] S. Gaikwad, S. S. Sawant, and J. S. Shastri, "Comparison of nonstructural protein-1 antigen detection by rapid and enzyme-linked immunosorbent assay test and its correlation with polymerase chain reaction for early diagnosis of dengue," *Journal of Laboratory Physicians*, vol. 9, no. 03, pp. 177–181, 2017.

[17] M. T. Nguyen, T. N. Ho, V. V. C. Nguyen, T. H. Nguyen, M. T. Ha, V. T. Ta, L. D. H. Nguyen, L. Phan, K. Q. Han, and T. H. K. e. a. Duong, "An evidence-based algorithm for early prognosis of severe dengue in the outpatient setting," *Clinical Infectious Diseases*, p. ciw863, 2016.

[18] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Predicting healthcare trajectories from medical records: A deep learning approach," *Journal of Biomedical Informatics*, vol. 69, pp. 218–229, 2017.

[19] A. Amidi and S. Amidi, "Cs 230 - recurrent neural networks cheatsheet." [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[20] M. Phi, "Illustrated guide to lstm's and gru's: A step by step explanation," 2018. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[21] 2021. [Online]. Available: https://www.cdc.gov/dengue/training/cme/ccm/page47831.html

[22] D. H. Ng, J. G. Wong, T.-L. Thein, Y.-S. Leo, and D. C. Lye, "The significance of prolonged and saddleback fever in hospitalised adult dengue," *PLOS ONE*, vol. 11, no. 12, p. e0167025, 2016.

[23] F. Chollet, *Deep learning with Python*. Manning, 2018.

Appendix A

Severe Dengue Predictors Result

	Accuracy		Precision		Sensitivity		Specificity		PPV		NPV	
	Training	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**
1-step-ahead	94 ± 3	95 ± 2	93 ± 3	96 ± 2	97 ± 3	96 ± 4	36 ± 3	94 ± 3	57 ± 2	95 ± 3	37 ± 2	95 ± 6
2-step-ahead	93 ± 4	95 ± 1	91 ± 4	93 ± 2	96 ± 6	98 ± 2	44 ± 3	92 ± 3	48 ± 3	91 ± 2	45 ± 2	98 ± 2
3-step-ahead	88 ± 4	89 ± 3	84 ± 4	86 ± 6	92 ± 8	92 ± 4	46 ± 3	90 ± 4	42 ± 3	85 ± 6	44 ± 3	92 ± 3
4-step-ahead	86 ± 4	86 ± 3	81 ± 4	84 ± 3	89 ± 8	84 ± 5	47 ± 3	87 ± 3	38 ± 4	81 ± 4	49 ± 1	86 ± 4
5-step-ahead	86 ± 4	86 ± 2	80 ± 4	82 ± 4	89 ± 9	85 ± 5	49 ± 3	87 ± 5	36 ± 4	80 ± 5	50 ± 1	88 ± 3
6-step-ahead	85 ± 4	85 ± 2	79 ± 4	81 ± 4	86 ± 8	82 ± 4	48 ± 3	86 ± 3	35 ± 3	78 ± 6	50 ± 2	86 ± 3
7-step-ahead	81 ± 3	79 ± 4	73 ± 3	72 ± 6	81 ± 10	73 ± 12	49 ± 3	82 ± 4	31 ± 4	70 ± 5	51 ± 2	82 ± 6
8-step-ahead	80 ± 3	83 ± 2	68 ± 4	75 ± 3	72 ± 12	75 ± 9	54 ± 3	82 ± 5	24 ± 4	65 ± 8	56 ± 2	84 ± 9
9-step-ahead	84 ± 3	85 ± 2	77 ± 7	83 ± 5	51 ± 11	53 ± 12	64 ± 4	89 ± 1	12 ± 3	77 ± 9	71 ± 2	86 ± 3

Table A.1: Full Performance of 9 Dynamic Severe Dengue Predictor trained with Maximum-data-length Format

	Accuracy		Precision		Sensitivity		Specificity		PPV		NPV	
	Training	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**
3-step-ahead	89 ± 4	90 ± 2	82 ± 4	85 ± 5	95 ± 3	96 ± 2	45 ± 3	89 ± 4	43 ± 1	84 ± 6	46 ± 3	95 ± 2
4-step-ahead	87 ± 4	88 ± 3	79 ± 4	85 ± 4	93 ± 3	87 ± 3	46 ± 3	87 ± 4	40 ± 1	82 ± 4	47 ± 3	88 ± 4
5-step-ahead	86 ± 4	87 ± 3	78 ± 4	81 ± 5	93 ± 4	91 ± 6	48 ± 4	85 ± 6	38 ± 2	78 ± 7	49 ± 3	91 ± 4
6-step-ahead	85 ± 4	85 ± 1	76 ± 4	79 ± 5	92 ± 3	87 ± 6	47 ± 3	85 ± 5	37 ± 1	77 ± 6	48 ± 3	89 ± 3
7-step-ahead	82 ± 3	80 ± 3	71 ± 4	70 ± 4	90 ± 5	84 ± 6	47 ± 3	79 ± 3	34 ± 2	67 ± 4	48 ± 2	87 ± 5
8-step-ahead	80 ± 3	81 ± 2	66 ± 4	69 ± 5	84 ± 5	82 ± 7	51 ± 4	82 ± 7	53 ± 2	62 ± 11	53 ± 2	90 ± 3
9-step-ahead	83 ± 3	86 ± 5	67 ± 6	73 ± 14	62 ± 9	75 ± 10	61 ± 4	86 ± 5	15 ± 2	67 ± 15	68 ± 2	92 ± 3

Table A.2: Full Performance of 9 Dynamic Severe Dengue Predictor trained with Maximum-data-length Format and $\alpha = 1.5$

	Accuracy		Precision		Sensitivity		Specificity		PPV		NPV	
	Training	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**
1-step-ahead	94 ± 3	95 ± 1	94 ± 3	97 ± 1	98 ± 1	97 ± 1	14 ± 2	79 ± 10	79 ± 1	97 ± 2	14 ± 3	70 ± 5
2-step-ahead	91 ± 5	94 ± 1	91 ± 4	95 ± 1	97 ± 2	97 ± 2	19 ± 3	85 ± 4	71 ± 1	95 ± 1	20 ± 4	94 ± 4
3-step-ahead	90 ± 5	93 ± 3	89 ± 5	93 ± 4	97 ± 2	92 ± 2	27 ± 3	84 ± 4	62 ± 1	92 ± 4	28 ± 4	93 ± 4
4-step-ahead	90 ± 5	92 ± 3	88 ± 5	91 ± 2	97 ± 2	97 ± 2	29 ± 3	85 ± 7	59 ± 1	90 ± 2	30 ± 4	93 ± 5
5-step-ahead	89 ± 5	91 ± 2	86 ± 4	88 ± 3	97 ± 2	99 ± 1	30 ± 3	83 ± 6	57 ± 1	86 ± 3	31 ± 4	88 ± 3
6-step-ahead	88 ± 4	89 ± 2	84 ± 4	83 ± 3	93 ± 3	97 ± 2	40 ± 3	82 ± 4	46 ± 2	85 ± 3	42 ± 3	96 ± 3
7-step-ahead	85 ± 4	85 ± 4	82 ± 4	81 ± 3	86 ± 3	88 ± 6	38 ± 3	78 ± 4	42 ± 2	81 ± 4	43 ± 3	88 ± 5
8-step-ahead	80 ± 4	83 ± 5	67 ± 8	70 ± 9	55 ± 18	67 ± 13	58 ± 4	87 ± 1	15 ± 5	66 ± 16	65 ± 3	88 ± 4
9-step-ahead	82 ± 3	81 ± 3	39 ± 34	36 ± 37	7 ± 9	15 ± 15	68 ± 5	88 ± 3	1 ± 2	26 ± 22	81 ± 3	36 ± 37

Table A.3: Full Performance of 9 Dynamic Severe Dengue Predictor trained with Median-data-length Format

Appendix B

Sample Data for Simulation

Patient ID	Day From Onset	Vomiting	Body Temperature	Respiratory Rate	Haemoglobin	Haematocrit Percent	Platelet Count	Bleeding Vaginal	Bleeding Mucosal	Abdominal Pain	Label
1888	0	1	0	0	0	0	0	0	1	0	1
1888	3	1	0	0	0.272727	38	53	0	1	0	
1888	4	1	0	0	0.272727	41.2	31.4	0	1	0	
1888	5	1	0	0	0.272727	46.7	21.1	0	1	0	
1888	6	1	0	0	0.272727	49	79	0	1	0	
1888	7	1	0	0	0.272727	48	102	0	1	0	
1888	8	1	0	0	0.272727	42	0	0	1	0	
1888	9	1	0	0	0.272727	40	0	0	1	0	
1888	10	1	0	0	0.272727	40	0	0	1	0	

Table B.1: Clinical Data of Patient 1888. Row 2 and 3 Were Used for Simulation

Patient ID	Day From Onset	Vomiting	Body Temperature	Respiratory Rate	Haemoglobin	Haematocrit Percent	Platelet Count	Bleeding Vaginal	Bleeding Mucosal	Abdominal Pain	Label
602	0	1	0	0	0	0	0	0	0	0	1
602	2	1	0	0	0	0	0	0	0	0	
602	4	1	39	18	0.351351	36.4	110	0	0	0	
602	5	1	0	0	0.351351	35.6	99	0	0	0	

Table B.2: Clinical Data of Patient 602. Row 3 and 4 Were Used for Simulation

Patient ID	Day From Onset	Vomiting	Body Temperature	Respiratory Rate	Haemoglobin	Haematocrit Percent	Platelet Count	Bleeding Vaginal	Bleeding Mucosal	Abdominal Pain	Label
641	0	1	0	0	0	0	0	0	0	0	0
641	2	1	0	0	0	0	0	0	0	0	
641	3	1	38	18	0	35.8	82	0	0	0	
641	4	1	0	0	0	39.7	98	0	0	0	
641	5	1	0	0	0	35.9	142	0	0	0	
641	6	1	0	0	0	35.5	146	0	0	0	

Table B.3: Clinical Data of Patient 641. Row 3 and 4 Were Used for Simulation

Patient ID	Day From Onset	Vomiting	Body Temperature	Respiratory Rate	Haemoglobin	Haematocrit Percent	Platelet Count	Bleeding Vaginal	Bleeding Mucosal	Abdominal Pain	Label
920	0	0	0	0	0	0	0	0	0	0	0
920	3	0	40	24	0.03	40.5	193	0	0	0	
920	4	0	0	0	0.03	41	166	0	0	0	
920	5	0	0	0	0.03	39	220	0	0	0	
920	6	0	0	0	0.03	39.9	187	0	0	0	
920	7	0	0	0	0.03	37.4	180	0	0	0	
920	8	0	0	0	0.03	41.2	202	0	0	0	

Table B.4: Clinical Data of Patient 920. Row 2 and 3 Were Used for Simulation

Patient ID	Day From Onset	Vomiting	Body Temperature	Respiratory Rate	Haemoglobin	Haematocrit Percent	Platelet Count	Bleeding Vaginal	Bleeding Mucosal	Abdominal Pain	Label
934	0	1	0	0	0	0	0	0	0	1	0
934	2	1	0	0	0.259459	36.5	136	0	0	1	
934	3	1	0	0	0.259459	32	148	0	0	1	
934	4	1	0	0	0.259459	36	60	0	0	1	
934	5	1	0	0	0.259459	33.7	93	0	0	1	
934	6	1	0	0	0.259459	39.5	100	0	0	1	
934	7	1	0	0	0.259459	46.6	50	0	0	1	
934	8	1	0	0	0.259459	34.5	64	0	0	1	
934	11	1	0	0	0	0	0	0	0	1	
934	118	1	0	0	0	0	0	0	0	1	
934	120	1	39	24	0	0	0	0	0	1	
934	122	1	0	0	0	0	0	0	0	1	

Table B.5: Clinical Data of Patient 934. Row 2 and 3 Were Used for Simulation