ORIGINAL ARTICLE

ARTIFICIAL INTELLIGENCE MODEL AS PREDICTOR FOR DENGUE OUTBREAKS

Dhesi Baha Raja[1¶], Rainier Mallol[2¶], Choo Yee Ting[3¶], Fadzilah Kamaludin[4], Rohani Ahmad[5], Suzilah Ismail[6], Vivek Jason Jayaraj[7*] and Bala Murali Sundram[8]

[1] Public Health Medicine Specialist, Ministry of Health, 62590 Putrajaya, Malaysia
[2]Computer Engineer, Pontifical Catholic University Mother and Master (Pontificia Universidad Católica Madre y Maestra), 51000 Santiago de los Caballeros, Dominican Republic
[3] Associate Professor, Faculty of Computing and Informatics, Multimedia University (MMU), 63100 Cyberjaya, Malaysia
[4] Director, Institute for Medical Research (IMR), Ministry of Health, 50588 Kuala Lumpur, Malaysia
[5] Medical Entomology Unit, Infectious Diseases Research Centre, Institute for Medical Research (IMR), Ministry of Health, 50588 Kuala Lumpur, Malaysia
[6] SQS Statistical Consulting, School of Quantitative Sciences(SQS), Universiti Utara Malaysia (UUM), 06010 Sintok, Kedah, Malaysia
[7] Department of Social and Preventive Medicine, University Malaya, 50603 Kuala Lumpur, Malaysia
[8]Public Health Medicine Specialist, Environmental Health Research Centre, Institute for Medical Research (IMR), Ministry of Health, 50588 Kuala Lumpur, Malaysia

***Corresponding author: Vivek Jason Jayaraj**
Email: vivekjason1987@gmail.com

Abstract

Dengue is an increasing threat in Malaysia, particularly in the more densely populated regions of the country. We present an Artificial Intelligence driven model in predicting Aedes outbreak, using predictors of weather variables and vector indices sourced from the Ministry of Health. Analysis and predictions to estimate Aedes populations were conducted, with its results being used to infer the possibility of dengue outbreaks at pre-determined localities around the Klang Valley, Malaysia. A Bayesian Network machine learning technique was employed, with the model being trained using predictor variables such as temperature, rainfall, date of onset and notification, and vector indices such as the Ae. albopictus count, Ae. aegypti count and larval count. The interfaces of the system were developed using the C# language for Server-side configuration and programming, and HTML, CSS and JavaScript for the Client Side programming. The model was then used to predict the population of Aedes at periods of 7, 14, and 30 days. Using the Bayesian Network technique utilising the above predictor variables we proposed a finalised model with predictive accuracy ranging from 79%-84%. This model was developed into a Graphical User Interface, which was purposed to assist and educate the general public of regions at risk of developing dengue outbreak. This remains a valuable case-study on the importance of public data in the context of combating a public health risk via the development of models for predicting outbreaks of dengue which will hopefully spur further sharing of data by all parties in combating public health threats.

Keywords: Aedes, Aegypti, Albopictus, C#, Bayesian Network, Dengue, Predictive Model

INTRODUCTION

The World Health Organization (WHO) recognises dengue as a major international public health concern, with severe dengue an important cause of morbidity and mortality in many Asian and Latin American countries(1). Dengue affects 2.5 billion people worldwide, with estimates placing 50-100 million of these infections occurring each year throughout the world(1). Dengue trend analysis indicate dengue cases are increasing at an explosive rate over the last several decades (1).

Current surveillance methods for dengue employed by agencies participating in dengue surveillance have been classified into either active, passive or sentinel site. The 2010 meeting between the Asia-Pacific Dengue Prevention Board (APDPB) and the Americas Dengue Prevention Board (ADPB) reported that passive surveillance of dengue is the most commonly practiced (76.9%), followed by sentinel site surveillance (19.2%) and finally, active surveillance being practiced least frequently (3.9%). The weaknesses of current surveillance are; preference by decision makers for dengue control over dengue prediction, lack of preventive dengue services, difficulty in interpretation of diagnostic tests, centralized reporting which prevents timely local actions, lack of virology surveillance, and lack of coordination and data sharing.

Alternative surveillance to traditional efforts have been attempted in recent times, via several domains, with an emphasis on detecting health-seeking behaviour during the early stages of dengue fever. Such efforts include the monitoring of telephone triage calls, school and/or work absenteeism, sales of over-the-counter drugs and even online activity(2). The results of such efforts

have been quite varied, in terms of correlation and timeliness.

One such method of alternative surveillance is that of online activity and behaviour monitoring. Google pioneered this effort through Google Flu Trends (2), whereby Google query data was used to estimate real-time influenza activity. The methodology used in Google Flu Trends was subsequently adapted for dengue surveillance (3), resulting in Google Dengue Trends (4). Other related mechanisms used data from Google Insights for Search (which was subsequently merged into Google Trends (5) which utilized Google search queries to predict dengue incidences in Singapore and Bangkok (6). Search query data has proven to be useful for conducting "now-casting" (7), to facilitate near real-time indication of dengue.

Predictors for the outbreak of dengue fever includes weather variability, mosquito density and imported cases, ovitrap surveillance, social-economic factors, population density and land use. Weather variability has both positive and negative effects on the spread of dengue. Rainfall (14) is positively correlated with the spread of dengue while temperature variation and relative humidity impede dengue spread (15). A recent study (16) on the effects of weather on dengue fever spread showed that temperature, precipitation, vapour pressure and minimum relative humidity was positively associated, whilst being negatively associated with air pressure. The same study also showed that mosquito density was a positive indicator, which validates the usage of ovitraps as a dengue indicator; as is shown by other related works (17–19). In addition to all these factors, factors such as human population density (18) and proximity to construction sites (18) are beginning to be investigated with regards to dengue fever spread. Socio-economic condition is another emerging area of research to be considered in the spread of dengue fever. The objective of our research is to develop an early forecasting model using variables such as temperature, rainfall, date of onset and date of notification, and vector indices such as the Ae. *albopictus* count, Ae. *aegypti* count and larval count using novel artificial intelligence methods. After which, our second objective would be to validate the particular model that has been developed.

METHODOLOGY

Predictor Data

The predictor variables used in training this model were temperature, rainfall, date of onset and date of notification, and vector indices such as the Ae. *albopictus* count, Ae. *aegypti* count and larval count from around the Klang Valley region over a period of 15 years, from 2003-2017. This data was acquired from the Institute for Medical Research, Malaysia and Ministry of Health Malaysia.

Bayesian Network Models for Aedes Population Prediction

An analytic project on *Aedes* population prediction was conducted to provide an estimation in inferring the possibility of dengue outbreaks at a specific location. In this project, a machine learning technique named Bayesian Network was deployed.

Formal Definition of Bayesian Networks

A Bayesian network B is an annotated acyclic graph that represents a Joint Probability Distribution (JPD) over a set of random variables ***V***. The network is defined by ***N*** = <***G***, ***P***>, where ***G*** is the DAG whose nodes X_1, X_2,...,X_n represents random variables, and whose edges represent the direct dependencies between these variables. The graph G encodes independence assumptions, by which each variable *X*i is independent of its non-descendent given its parents in ***G***. The second component ***P*** denotes the set of parameters of the network. This set contains the parameter for each realization x_i of Xi conditioned on πi, the set of parents of Xi in ***G***. Accordingly, ***N*** defines a unique JPD over ***V***, namely:

$$\vartheta_{xi \mid \pi i} = P_{B(xi \mid \pi i)}$$

$$P_B(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P_B(X_i \mid \pi_i = \prod_{i=1}^{n} {}^{\theta} X_i \mid \pi_i$$

If Xi has no parents, its local probability distribution is said to be unconditional, otherwise it is conditional. If the variable represented by a node is *observed*, then the node is said to be an evidence node, otherwise the node is said to be hidden or latent. The inference mechanism in Bayesian Networks is not straight forward, mainly because propagation of information can happen in bi-directionally, with or against the arc direction. In addition, the property of *d-separation* captures both the conditional independence and dependence relations that are implied by the Markov condition on the random variables.

Bayesian Network Learning

Bayesian network learning is practical in real-world, particularly when the causal dependencies between the variables are not explicitly known by humans. There are two types of learnings: (1) parameter learning and (2) structure learning. In parameter learning, the goal of learning is to find the values of the BN parameters (in each CPD) that maximize the (log)likelihood of the training. This dataset contains m cases that are often assumed to be independent. Upon training dataset $T = \{x_1, ..., x_m\}$, where $x_{|} = x_{|1}, ..., x_{|n})^T$, and the parameter set $Q = \vartheta_1, ..., \vartheta_n)$ where θi is the vector of parameters for the conditional distribution of variable Xi (represented by one node in the

graph), the log-likelihood of the training dataset is the sum of terms, one for each node:

$$\log L(\Theta \mid \Sigma) = \sum_m \sum_n \log P(x_{li} \mid \pi_i, \theta_i)$$

User Interface Development

The interfaces of the system were developed using the C# language for Server-Side configuration and programming and, HTML, CSS and JavaScript for the Client-Side programming.

Client-Side Environment

The client-side environment used to run scripts for this application lies in the user's browser. The processing takes place on the end users computer. The source code is transferred from the web server to the user's computer over the internet and runs directly in the user's browser. The scripting language needs to be enabled on the client computer. In modern browsers, this configuration is usually set by default.

Server-Side Environment

The server-side environment running the scripting language is a web server. For this application, we have used web servers hosted in the cloud, namely the Azure cloud services from Microsoft. A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser. It is usually used to provide interactive web sites that interface to databases or other data stores on the server.

This is different from client-side scripting where scripts are run by the viewing web browser, usually in JavaScript. The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores.

Designing the Interface

The interface was designed with the following intentions:

- Effective: It accomplishes the task of providing information related to predictions, versus the cases that happened days after the prediction.
- Speed: Users can easily and quickly understand the information we are trying to show.
- Portability: It can be used on any device, without sacrificing the above-mentioned intentions.

The interface used is mainly a horizontal interface, to provide the user with a macro view of the Maps with the cases and predictions, with a second half of the screen, visualizing the value and information of each prediction.

To create the interface, we used a CSS-JS framework called Bootstrap. Bootstrap is modular and consists of a series of *Less Stylesheets* that implement the various components of the toolkit. These stylesheets are generally compiled into a bundle and included in web pages, but individual components can be included or removed. Bootstrap provides a number of configuration variables that control things such as color and padding of various components.

Each Bootstrap component consists of a HTML structure, CSS declarations, and in some cases accompanying JavaScript code. Grid system and responsive design comes standard with an 1170 pixels wide grid layout. Alternatively, the developer can use a variable-width layout. For both cases, the toolkit has four variations to make use of different resolutions and types of devices: mobile phones, portrait and landscape, tablets and PCs with low and high resolution. Each variation adjusts the width of the columns.

Stylesheets

Bootstrap provides a set of stylesheets that provide basic style definitions for all key HTML components. These provide a uniform, modern appearance for formatting text, tables and form elements.

Java Script Components

In addition to the regular HTML elements, Bootstrap contains other commonly used interface elements. The components are implemented as CSS classes, which must be applied to certain HTML elements in a page.

Bootstrap comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields. In version 2.0, the following JavaScript plugins are supported: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel and Typeahead.

In order to visualize the prediction, the platform requires input from the user: user selects a timeline for prediction and user selects a location for prediction

Ethics approval and consent to participate not applicable. This is a project conducted using an ecological outlook utilising count data of cases over a period of weeks with no personal identifiers of patients.

RESULTS

Developing the Predictive Model using Bayesian Networks

The model developed using Bayesian networks integrated 7 variables using which it predicted the volume of Aedes that was possible in a said

locality. We assume here the magnitude of predicted Aedes population would be analogous to dengue incidence as more mosquitoes.
Figure 3.1,Figure 3.2,and Figure 3.3 depict the next 7 days, Bayesian Network models for *aegypti*, *albopictus*, and larval counts, respectively. As depicted in the figures, the nodes of interest are those in yellow. These models take the dataset in the following structure: D = {D_T, D_R, D_{aegy}, D_{albo}, D_{notify}, D_{onset}, D_{larva}, next7Days}

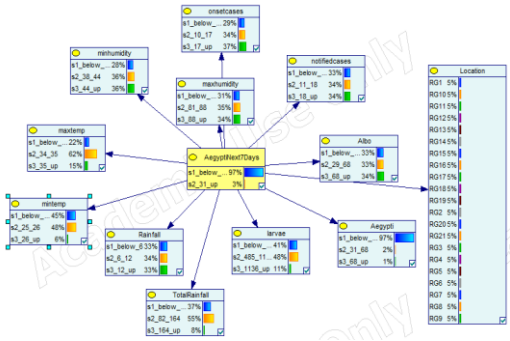

Figure 0.1 Aegypti Next 7 Days Bayesian Network Model

Figure 3.1, Figure 3.2, and Figure 3.3 above are the data structure representing utilised the dataset required for training the model- which comprises of temperature data, rainfall data, onset of cases, notification of cases, *albopictus* count, *aegypti* count and larval count, to predict the population of *Aedes* in the next 7 days. The Bayesian Network models can then be extended to predict populations over the next 14, and 30 days.

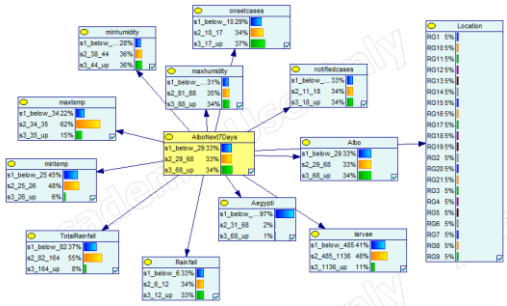

Figure 0.2 Albopictus Next 7 Days Bayesian

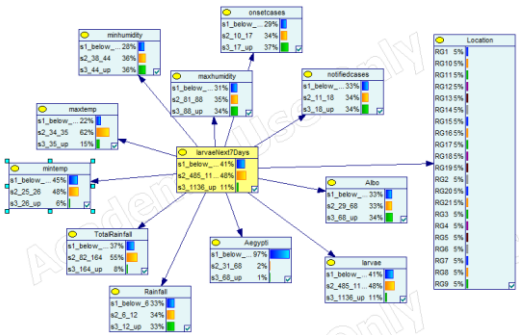

Figure 0.3 Larval Count Next 7 Days Bayesian Network

Merging Predictive Model to User Interface
In order to give valid options for the user to answer, we analyzed the data used to create the model, and developed two dropdowns with all of the possible options. To be able to create the dropdowns, we used an IEnumerable <> class. IEnumerable is the base interface for all non-generic collections that can be enumerated. The generic version of this interface see System. Collections. Generic. IEnumerable <T>. IEnumerable contains a single method, Get Enumerator, which returns an IEnumerator. IEnumerator provides the ability to iterate through the collection by exposing a current property using Move Next and Reset methods.

It is a best practice to implement I Enumerable and I Enumerator on your collection classes to enable the foreach (For Each in Visual Basic) syntax, however implementing I Enumerable is not required. If your collection does not implement I Enumerable, you must still follow the iterator pattern to support this syntax by providing a Get Enumerator method that returns an interface, class or struct. When using Visual Basic, you must provide an I Enumerator implementation, which is returned by Get Enumerator. When developing with C# you must provide a class that contains a Current property, and Move Next and Reset methods as described by I Enumerator, but the class does not have to implement I Enumerator.

After the dropdowns are created, we create a POST request containing the data to be sent. By design, the POST request method requests that a web server accept the data enclosed in the body of the request message, most likely for storing it. It is often used when uploading a file or when submitting a completed web form. In contrast, the HTTP GET request method retrieves information from the server.

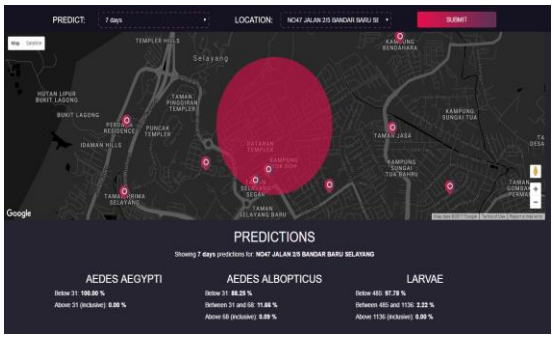

Figure 0.4 User interface example 1

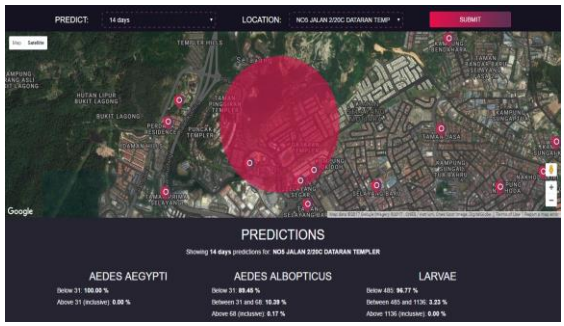

Figure 0.5 User interface example 2

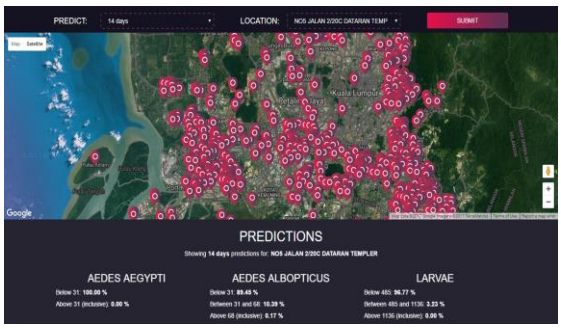

Figure 0.6 User interface example 3

The POST request reaches the server, through a C# action method or MVC controller. C# MVC Controllers are responsible for controlling the flow of the application execution. When you make a request (means request a page) to MVC applications, a controller is responsible for returning the response to that request. A controller can have one or more actions.

Once the data is received, the server uses the predictive model, by inputting into it the data that has been sent from the user. Once the data is in, the model returns several predictions. These new sets of data are sent to the Client side (the browser) of the user.

Finally, once the data is in the browser of the user, we use several mapping mechanisms to create a map, give it style (size and colors), and draw all of the needed shapes. The end product of the interface and predictions can be seen in Figure 3.4, Figure 3.5, and Figure 3.6, below.

DISCUSSION

The results of this work showed that the use of entomologic data can be useful in modelling a dengue outbreak by using meteorological, epidemiological, and geo-social data as predictors. The model shows high accuracy on metrics.

Our results compared favourably with other similar work. Rainfall and temperature was used as an indicator for an *Aedes* outbreak in India, using a Neural Network model. The accuracy for this work ranged from 81% to 92%, depending on the number of hidden nodes. In Peru, indicators used for the prediction of dengue outbreaks included rainfall, temperature, vegetation cover, atmospheric pressure, sea surface temperature, elevation, as well as socio-economic and demographic data. The fuzzy-association rules based model used for this work resulted in near 100% specificity, although sensitivity was reported as about 60%. Our model outperformed the multiple rule-based model which was used to predict dengue outbreak in Malaysia, with an AUC of 73.78 produced by the Naïve Bayes classifier.

While the results validated the use of entomology data, there are significant issues and challenges that need to be addressed in order to improve the usability of these data, as predicting dengue outbreaks with entomology data in a machine learning engine is a novel approach.

The second challenge is that of timeliness of the data provided. While census data is periodic by nature, entomology and meteorological data requires a significant turn-around time before it is provided on demand. This is a significant challenge for any effort to implement near-real-time monitoring of dengue.

A third challenge is the uniformity of data format and data sources. Data sharing has begun for the dengue cases via official open data portal of e-dengue as a spreadsheet, portable document format, or as a graphic file while entomology data are in excel sheets. Adoption of standard data definition, documents, and interoperability are steps needed to encourage the use of these data.

The study has also to a large extent highlighted the need for open data solutions to be implemented within health care systems. Health related Open Data has seen increased usage over the years. Epivue (8), an open source public health information system was used to manage and disseminate public health data from the Washington State Cancer Registry and Washington State Centre for Health Statistics. HealthData.gov has published 400 health datasets (9), which contain data ranging from epidemiology to MediCare data.

In the UK, open health data has been used to potentially lower the cost of statins prescribed under the United Kingdom's National Health Service (10) by examining a public health data set containing prescriptions written by every family physician. Health related Open Data can result in innovative solutions, as demonstrated during the Health 2.0 Developer Challenge (11) organized by the New York State (NYS) health department. In the challenge, NYS made data on doctors and hospitals in New York state available to developers, purposed towards developing applications which can help members of the public make more informed health decisions. Open Data is also seeing improving adoption in under-developed nations in Africa. In Uganda, the current status of health services in three districts have been made available to non-governmental organizations (NGO) and the mass media for dissemination to the wider public (12).

In Malaysia, open data for health is limited to press statements and press releases to the general public issued by the Ministry of Health and its officers, and also by statements made by the Minister of Health. A recent effort at using from the Ministry of Health can be found in iDengue

(13), an informational website which lists out number of cases in Malaysia and their corresponding geo-location. iDengue does feature pattern recognition, which is addressed in our work. Increasing health related open data sources particularly utilising data via artificial intelligence models such as those proposed here can also expand the utility of data beyond its current use in information, quality and operational systems.

CONCLUSION

We have contributed a model for predicting outbreaks of dengue using entomology data, driven by artificial intelligence. This model performs well and compares favourably with other similarly efforts. We hope that the success of creating this model will encourage further sharing of data by all parties, especially when the data is to be used to combat a public health threat.

ACKNOWLEDGEMENTS

The authors would like to thank the Director General of Health Malaysia for his permission to publish this article.

REFERENCES

1. El Adlouni S, Beaulieu C, Ouarda TB, Gosselin PL, Saint-Hilaire A. Vector Borne Disease. Int J Health Geogr. 2007;6(1):40.

2. Brady OJ, Smith DL, Scott TW, Hay SI. Dengue disease outbreak definitions are implicitly variable. Epidemics. 2015 Jun;11:92–102.

3. Packierisamy PR, Ng C-W, Dahlui M, Inbaraj J, Balan VK, Halasa YA, et al. Cost of Dengue Vector Control Activities in Malaysia. Am J Trop Med Hyg. 2015 Nov;93(5):1020–7.

4. Chaudhry B, Wang J, Wu S, Maglione M, Mojica W, Roth E, et al. Systematic Review: Impact of Health Information Technology on Quality, Efficiency, and Costs of Medical Care. Ann Intern Med. 2006 May 16;144(10):742.

5. Drury P. The eHealth agenda for developing countries. World Hosp Health Serv. 2005;41(4):38–40.

6. Delone WH, Mclean ER. Information Systems Success: The Quest for the Dependent Variable. MIS Q. 1988;12:259–74.

7. Doll WJ, Torkzadeh G. The Measurement of End-User Computing Satisfaction: Theoretical and Methodological Issues. MIS Q. 1991 Mar;15(1):5.

8. Davis FD. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Q. 1989 Sep;13(3):319.

9. Zaichkowsky JL. Measuring the Involvement Construct. J Consum Res. 1985 Dec 1;12(3):341.

10. Myers MD. Qualitative Research in Information Systems. MIS Q. 1997 Jun;21(2):241.

11. Gable GG, Sedera D, Chan T. Enterprise Systems Success: A Measurement Model. Proc Twenty-Fourth Int Conf Inf Syst. 2003;576:145–76.

12. Sedera D, Gable G. A factor and structural equation analysis of the enterprise systems success measurement model. Int Conf Inf Syst. 2004;449–64.

13. Seddon PB, Graeser V, Willcocks LP. Measuring organizational Information System effectiveness: An Overview and Update of senior management perspectives. ACM SIGMIS Database. 2002 Jun 1;33(2):11–28.

14. Hair JF, Black WC, Babin BJ, Anderson RE. Multivariate data analysis - Pearson. Pearson Education Inc; 2010.

15. Seddon P, Kiew M-Y. A Partial Test and Development of Delone and Mclean's Model of IS Success. Australas J Inf Syst. 1996 Nov 1;4(1).

16. Venkatesh V, Morris MG, Davis GB, Davis FD. User Acceptance of Information Technology: Toward a Unified View. Source MIS Q. 2003;27(3):425–78.

17. Wyatt J. Clinical data systems : Development and evaluation. Lancet. 1994 Dec 17;344(8938):1682–8.

18. Rockart JF (John F, DeLong DW. Executive support systems : the emergence of top management computer use. Dow Jones-Irwin; 1988. 280 p.

19. Koivunen M, Välimäki M, Koskinen A, Staggers N, Katajisto J. The impact of individual factors on healthcare staff's computer use in psychiatric hospitals. J Clin Nurs. 2009 Apr 1;18(8):1141–50.