

## **Design of a low cost and novel Naive Bayes classifier based NodeMCU web server for fever type detection.**

**Partha Pratim Ray<sup>1\*</sup>, Dinesh Dash<sup>2</sup>, Debashis De<sup>3</sup>**

<sup>1</sup>Department of Computer Applications, Sikkim University, Gangtok, Sikkim, India

<sup>2</sup>Department of Computer Science and Engineering, National Institute of Technology, Patna, India

<sup>3</sup>Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology (formerly West Bengal University of Technology), Kolkata, India

### **Abstract**

**This study presents a novel NodeMCU based web server to validate cost-effective and smart e-health diagnosis. Proposed work implements a novel Naïve Bayes Classifier for automatic fever type detection as a proof of concept. The application is currently accessible on local network. Network users (patients) are first required to submit eight symptoms e.g. “fatigue”, “fever”, “chills”, “sore throat”, “cough”, “headache”, “muscle pain”, and “sneezing” on web portal in form of yes “y” or no “n”. Implied Naïve Bayes Classifier engine predicts the probability of occurrence of fever either from flu or common cold as per symptoms provided earlier. The patients are simultaneously diagnosed by a medical practitioner from whom patient wise predictive percentage was received. The probabilistic values were then paired against flu and common cold typed fever patients independently. 22 patients were voluntarily gone through this experiment where  $p(0.089 > 0.05)$  and  $p(0.068 > 0.05)$  were not found to be statistically significant (i.e. no difference between proposed classifier and doctor’s diagnosis) for flu and common cold fever types, respectively.**

**Keywords:** E-health, Web server, Naïve Bayes classifier, Fever detection.

*Accepted on January 22, 2018*

### **Introduction**

A fever is an unusual state of body temporary ordinarily measured higher than the normal range (37.5 and 38.3°C) [1]. Fever can be caused due to any medical irregularities such as bacterial, parasitic, viral infections or any such non-infectious causes [2]. It is comprehended as the most common medical symptom among 30% and 70% of seriously ill children and elders, respectively [3]. Though, it is an auto-reflexive outburst of bodily immune-mechanism, consequences of fever have seen as a great concern to a large set of parents and medical practitioners world-wide [4,5]. Infectious causes such as flu (influenza) and common cold have severe impressions on it. According to the cold and flu statistics [6], an adult usually gets common cold 2-4 times a year, whereas 5-10 times for a child. Alone, flu causes 3-5 million hospitalizations by incurring 250,000-500,000 death trolls each year [7]. More than 10 billion USD expenditure are accounted per year against related treatments [8,9]. Thus, relevant contributions are necessary to minimize the inherent risks (e.g. life and cost) associated with fever especially in common cold and flu. E-health centric outcome could be most suitable in this aspect due to user friendliness, ease of accessibility and cost effectiveness.

This paper presents a novel and low cost Artificial Intelligence (AI) based approach towards solving this issue. In this experiment, an open source-Internet of Things (IoT) [10] hardware platform-NodeMCU is used to cater all computational and decision-making process. NodeMCU runs on ESP8266 Wi-Fi System on Chip (SoC) that has 128 kB RAM and 4 MB data storage capacity. NodeMCU is designed to best suited for resource constrained environment where by leveraging complete autonomy in term of control and computation in a highly-distributed network paradigm. An integrated Naïve Bayes Classifier (NBC) engine has provisioned valuable decision-making process in this regard. The main objective of this work is as follows:

1. To design a cost-effective and novel e-health as a proof of concept web server.
2. To implement Naïve Bayes classifier algorithm in NodeMCU storage facility.
3. To design a Graphical User Interface (GUI) for easy access to users.
4. To validate efficiency of Naïve Bayes classifier towards fever type detection with actual diagnosis.

## Materials and Methods

### Materials used

This study involves two main components such as (1) NodeMCU; (2) PC. As stated earlier, NodeMCU is a 5 USD hardware platform to facilitate user’s query over IEEE 802.11 b/g/n Wi-Fi connectivity. Thus, it has promising nature to act as the resource-constrained network gateway. Besides that, it comprises of an L106 32-bit RISC microprocessor core running at 80 MHz clock frequency. Further, 16 GPIO pins, SPI, I2C, I2S, UART, and 10-bit ADC are also integrated with it. Moreover, the Lua-scripting firmware provides descent flexibility over programing and debugging instructions. Subsequently, a Dell Inspiron model with 8 GB RAM, 1 TB HDD is used to burn the algorithm into the USB powered

NodeMCU. Arduino IDE (version 1.8.1) has been used as algorithm development platform on local network.

### Design methodology

Acting of employed NodeMCU depends on two different algorithmic attributes that includes: (1) Naïve Bayes Classifier engine; (2) Inherent web server activity. Figure 1 presents system model of proposed solution. NodeMCU is the core of the system where NBC algorithm and web server algorithm are juxtaposed together. A supervised learning engine controls the flow of NBC toward achieving correct results. The system works on a local network whether NodeMCU is placed at a distant location (to resemble with distributed approach). User on same network can access this NBC engine *via* TCP/IP protocol stack.

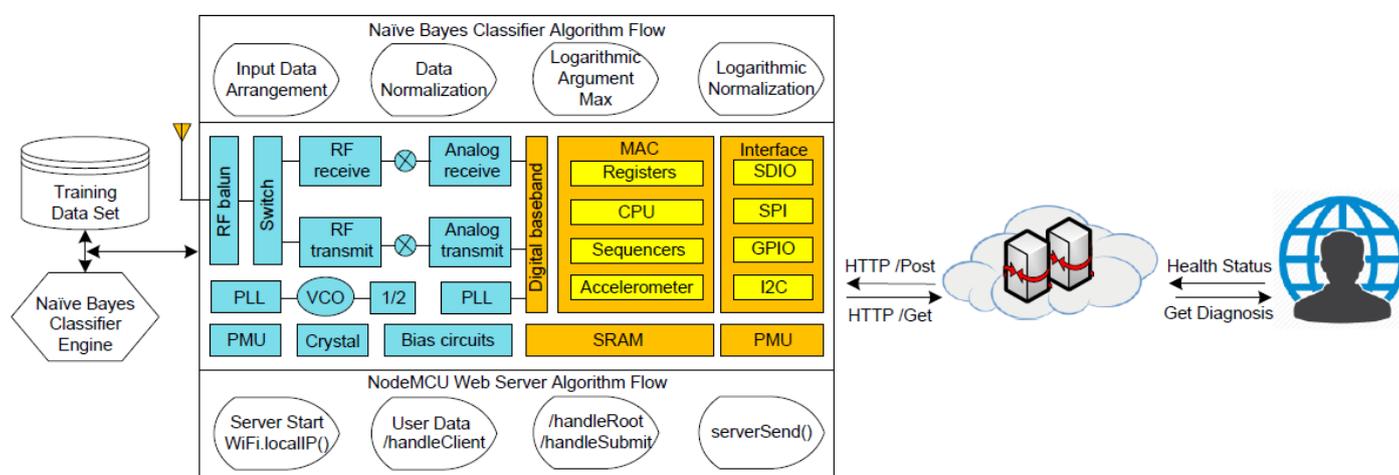


Figure 1. System model of Naive Bayes classifier based fever detection web server.

User provides 8 health status indicator values in yes (“y”) or no (“n”) text valued form *via* HTTP/Post to the web server. Upon successful reception of user’s symptom related information, NBC engine performs a supervised learning technique to retrieve appropriate type of fever (i.e. common cold or flu). The diagnosed results are then forwarded to the user *via* HTTP/Get method call. Detailed discussions on theoretical aspects and underlying assumptions are prescribed below.

### Naïve Bayes classifier

It is a well-known classifier algorithm that is commonly seen in automatic medical diagnosis [11]. The reason behind such intervention is its super scalability that requires numerous parameters linear with respect to feature/predictor labelling under a supervised learning environment where maximum-likelihood training is given per closed-form evaluation training. The Naïve Bayes Probabilistic model can be assumed as equation (1):

$$p(C_k | x_1, \dots, x_n) = (1/Z) p(C_k) \prod_{i=1}^n p(x_i | C_k) \text{---(1)}$$

Where Z is evidence i.e. equal to:  $p(x)$  that is independent of vector  $x (x_1, \dots, x_n)$ , where n is some features, k is possible outcome and  $C_k$  is set of classes. The resultant classifier is

modelled per Maximum a Posteriori (MAP) decision rule as given in equation (2):

$$\hat{y} = \text{argmax}_k p(C_k) \prod_{i=1}^n p(x_i | C_k) \text{---(2)}$$

Where  $k \in \{1, \dots, k\}$  and class label  $\hat{y} = C_k$

### Common cold and Flu classifier model

As mentioned earlier, a person gets affected by common cold 3-15 times a year. Assuming total population on earth 7.4 billion and average 9 times (i.e. (3+15)/2) cold, total common cold cases per year could be estimated as 66.6 billion. Similarly, around 5 million populations get flu ever year. Upon calculation of equations (3) and (4), it is found that 1 out of approx. 14,000 common cold/flu cases are actually flu.

$$p(\text{Flu}) = (5 \times 10^6) / (5 \times 10^6 + 66.6 \times 10^9) \approx 0.00007 \text{---(3)}$$

$$p(\text{Cold}) = (66.6 \times 10^9) / (5 \times 10^6 + 66.6 \times 10^9) \approx 0.93016 \text{---(4)}$$

It was also found that knowledge about high rise of body temperature would not sufficiently classify a type of fever than the other, as body temperature may not be directly correlated with cold or flu infection. Hence, in this experiment, eight more symptoms have been considered for more accurate classification results that includes “fatigue”, “fever”, “chills”,

“sore throat”, “cough”, “headache”, “muscle pain”, and “sneezing”. The resultant probability lookup database was retrieved from [9].

**Algorithm design**

As stated earlier, two separate algorithms are integrated together in NodeMCU. Firstly, the NBC algorithm; this algorithm performs the decision making task based on user’s symptoms. It first arranges the user’s symptom data per a Boolean nomenclature. Then, the data are normalized according to byte structured probabilistic model. Logarithmic argument “MAP-ing” is done thereafter. It is worth to note that, logarithmic MAP is important in this scenario. Because, NodeMCU is based on an embedded SoC that inherently lacks in floating point computational capacity. Logarithmic scaling has uplifted the efficiency of computations obligations in this regard. Lastly, the computed data are exported to user in form of strings whereby evaluating per exponential scale priory.

**Table 1.** Paired T-test values.

Group	Flu fever		Common cold fever	
	NBC	Doctor	NBC	Doctor
Mean	7.90	8.40	8.559	8.682
SD	1.432	1.14	0.864	0.81
SEM	0.64	0.51	0.21	0.196
N	5	5	17	17
Pearson’s r	0.9495		0.9534	
95% CI	6.122-9.678	6.98-9.82	8.115-9.003	8.266-9.099
t	2.24		1.95	
P value	0.089>0.05		0.068>0.05	
Significance	Not significant		Not significant	

Second algorithm is developed to run NodeMCU upon TCP/IP layer. The reason why TCP/IP is used in this experiment is its predominant nature of serving client/server network architecture which is implemented here. Initially, the server is set at 115200 baud rate for high speed “text” data communication over local network. It then scans for available local Wi-Fi network. Upon establishment of successful connection with designated SSID, NodeMCU returns a maskable and dynamic IP address on which the communication between user and NBC engine relies. Web server continuously looks upon available client connectivity over the network. If it perceives any such request from client (i.e. user), it promptly activates root handling functionality to correlate with client’s submissions. All “symptomatic” submissions are hereby checked for integrity and passed over to submit handling function. Here, server checks for “argument” mapping with predefined set of parametric argument list. If successful, then it transfers the activity linkage toward NBC engine where diagnosis takes place. The diagnosed report is then sent to client for his/her knowledge about particular type of fever

he/she is being suffering from. At this point, it is worth to note that the series of conversion of data string (e.g. from string to Boolean, to byte, to float, to string again) has played a crucial role in this experiment (Table 1).

**Results and Discussion**

We enquired 22 unique and independent participants who voluntarily replied against the symptoms as a probable case of flu or cold fever. They were simultaneously diagnosed by a medical practitioner from whom we received percentage wise score as proof of doctor’s diagnostic confidence. Same values were later tested against the NBC’s percentage wise diagnostic prediction values upon submission of each participant’s symptom into NodeMCU based NBC engine. A two-tailed paired T-test was performed independently for each of the set of patients (e.g. group of flu fever *versus* doctor and group of cold fever *versus* doctor). Hypotheses were assumed as follows.

H<sub>0</sub>: There was no difference between NBC and doctor’s prediction for each type of fever;

H<sub>A</sub>: There was difference between NBC and doctor’s prediction for each type of fever.

5 and 17 participants were screened for flu and common cold fever, respectively. The hypothesis was first tested on applicability of NBC for flu fever against 95% confidence interval, where we found P=0.089>0.05 i.e. difference was considered not to be statistically significant. Thus, the null hypothesis for diagnosis of flu cannot be rejected. The SEM was found 0.64 and 0.51 for NBC and doctor diagnosis, respectively. In second experiment, NBC was applied against diagnosis of common cold fever against 95% confidence interval. The result showed P=0.068>0.05 was not found to be statistically significant. Hence, the null hypothesis for measuring efficiency of NBC to detect common cold cannot be rejected. The SEM was found 0.21 for NCB and 0.196 doctor’s diagnosis in this case, respectively.

This paper presents a novel embedded web server based NBC predictor system for detection of cold/flu fever type. However, the system has limitation that includes network scalability. Current system was developed under local network scenario as a proof of concept. Hence, cloud and IoT related issues are missing that would be more beneficial for e-health and telemedicine services. In our future work, we will concentrate to develop a low cost, interoperable, mobile and wearable web server for such type of disease classification in more complex form.

**Conflict of Interest**

The authors declare no conflict of interest.

**References**

1. Axelrod YK, Diringer MN. Temperature management in acute neurologic disorders. *Neurol Clin* 2008; 26: 585-603.

2. Garmel. An introduction to clinical emergency medicine (2nd Ed.). Cambridge Univ Press 2012; 5.
3. Kiekkas P, Aretha D, Bakalis N, Karpouhisi I, Marneras C, Baltopoulos GI. Fever effects and treatment in critical care: literature review. Australian Critical Care 2013; 26: 130-135.
4. Niven DJ, Stelfox HT, Laupland KB. Antipyretic therapy in febrile critically ill adults: A systematic review and meta-analysis. J Critical Care 2013; 28: 303-310.
5. Schaffner A. Fever--useful or noxious symptom that should be treated? Therapeutische Umschau Revue therapeutique 2006; 63: 185-188.
6. Health Direct. Colds and flu statistics. Health Direct 2017.
7. WHO. Influenza (Seasonal) report. World Health Organization 2017.
8. WebMD. What are your odds of getting the flu? WebMD 2017.
9. Bindhammer M. Medical tricorder-Using artificial intelligence to identify a disease by its symptoms. HACKADAY 2017.
10. Ray PP. A survey of IoT cloud platforms. Future Computing Inform J 2017.
11. Rish, Irina. An empirical study of the naive Bayes classifier (PDF). IJCAI Workshop on Empirical Methods in AI 2001.

**\*Correspondence to**

Partha Pratim Ray

Department of Computer Applications

Sikkim University

India