International Journal on Emerging Technologies 11(1): 257-262(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

A Machine Learning Approach of Data Mining in Agriculture 4.0

Mahendra Swain[1], Rajesh Singh[2], Amit Kumar Thakur[3] and Anita Gehlot[4]
[1]Ph.D. Scholar, Department of Electronics Engineering, Lovely Professional University, Phagwara (Punjab), India.
[2]Professor, Department of Electronics Engineering, Lovely Professional University, Phagwara (Punjab), India.
[3]Associate Professor, Department of Mechanical Engineering,
Lovely Professional University, Phagwara (Punjab), India.
[4]Associate Professor, Department of Electronics Engineering,
Lovely Professional University, Phagwara (Punjab), India.

(Corresponding author: Amit Kumar Thakur)
(Received 11 October 2019, Revised 13 December 2019, Accepted 23 December 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: In the present era of internet, data mining is one of the most important features of all embedded systems. To fulfil the current food demand in India farmers have to adapt new technologies. In context to agriculture, crop species and yield prediction are very crucial to measure. The article explains how to incorporate machine learning algorithm to embedded hardware platform for data analysis to assist farmers to go for precision farming. The prototype has been designed using Raspberry Pi. Machine learning algorithms has been implanted for data acquisition from various sensor like pH sensor, flame sensor, pressure sensor etc. This study examines SVM implementation in embedded processor architectures and offers improved architectural efficiency. The CAN Protocol is used to collect and relay data from different sensors to the learning algorithm of the computer. TensorFlow software has been embedded to raspberry pi to implement machine learning. CAN is an asynchronous serial communication multi-master, multi-slave protocol mainly used in automobiles for In-vehicle communication. It works on the basis of node message priority instead of address priority, so there is no problem in adding or removing a node in already existing network. CAN provide maximum 1Mbps of data transfer rate, which is more than sufficient to send sensor data from one node to another node. CAN is highly robust and fault tolerant for errors. In this paper it briefly explains how Raspberry pi with machine learning algorithm could be used for big data problem. The prototype can be implemented in Smart cities, smart home, computational biology, energy production etc.

Keywords: SVM, CAN, Kernel, Optimization, Raspberry pi, Machine learning, Flame sensors.

Abbreviations: CAN, controller area network; SVM, support vector machine; GPIO, general purpose input output; HDMI, high-definition multimedia interface; GPU, graphics processing unit; UART (universal asynchronous receiver-transmitter), TensorFlow.

I. INTRODUCTION

Learning from past experience is one of Machine Learning's key attributes. As a result, developments in learning process theory and computer modeling are of great importance in areas associated with intelligence comprehension. Machine learning algorithms have been adapted to Embedded Systems for overall performance enhancement. Support vector machines (SVMs), because their mathematical background is based on the theory of statistical learning [2], are considered to be mainly commanding classification engines. SVMs have therefore shown a high degree of commitment in the area of SVM hardware design to help fill the gap between real-time performance and high detection accuracy. Uses of this technology in agriculture could bring potential revolution [16]. CAN (Controller Area Network) has been around 15 years and has not achieved its peak. The number of CAN nodes sold and built continues to increase year after year and is presently more than 200 million new nodes per year. This is mainly used for high speed application. ISO11519 developed CAN standard which explains about low speed applications up to 125kbps. All three standards use differential electrical voltage and the bus as a physical interface [5]. CAN bus uses only two wires as a physical interface as in Fig. 1. This is mainly used to reduce the bulky wiring harness in a vehicle system. Design transparency has been achieved, CAN have been designed on the basis of OSI reference model. In CAN, hardware physical layer and Data link layer have been implemented internally. Remaining layers of OSI model have been designed and into account by design engineers and higher level protocols [4]. CAN is a two wire twisted pair cable which is connecting all the nodes. The operation of CAN bus is simply a wired AND logic [1]. Always dominant bit will get priority. One wire is called CAN high and another is CAN low. CAN bus uses differential voltage level as an electrical signal to transmit a message on a CAN bus.

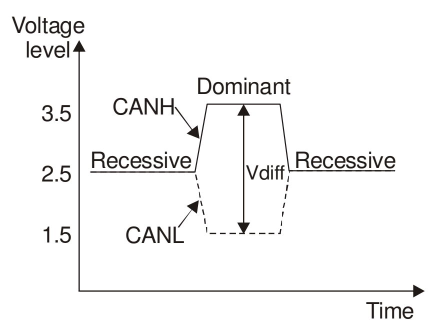

Fig. 1. CAN Bus.

CAN is working on the basis of message priority instead of node address priority so without changing hardware or software we can easily add or remove number of nodes

in CAN network. So CAN is highly flexible. It is reduced by posting a node to disable mode whenever it is required. CAN is a fixed priority arbitration mechanism.
Priority is decided by single unique identifier of that particular message. Every message's priority on a CAN bus is set ID doesn't indicate destination of a message instead it describes meaning of data and shows the importance of a message it is currently transmitting. Lower the ID higher the priority. If ID is high then message priority is less. Whether to accept or reject a message depends on the basis of message identifier in message filtering process of receiver side.
Number of nodes connected in CAN bus theoretically has no limit. The node count depends only on delay line or electrical load on CAN bus. In CAN bus every transmission of message on CAN bus ACK will be send to transmitter node by the receiver node for consistent delivery of message. Flag will be set by the CAN hardware for every inconsistent message delivery to receiver node. This bit should be cleared by programmer for reception of new message on CAN bus or retransmission of lost message. CAN supports multicasting, at the same time all nodes attached with CAN is able to receive message and accordingly it will act upon it. It is receiver nodes choice whether to accept or discard the correctly received message. CAN provide maximum safety for data transfer. Each node carries out signalling and self-checking.
The Raspberry Pi was created by the Raspberry Pi Foundation in the UK, and is a single-board computerIt consists of BCM2835 Broadcom chip (SoC), comprising an ARM1176JZF-S 700 MHz, Video Core IV graphics processing unit (GPU) originally intended for set-top box and with much more different field applications. It is a cost effective and multi purposed device. It operates on 700 MHz frequency, with 512 MB of RAM (in Model B). Raspberrypi Model B supports two USB ports, Ethernet port, Audio jack 3.5 mm and SD card slot for installation with any supported OS. It also supports Ethernet and High-Definition Multimedia Interface (HDMI) ports. The 26 GPIO (general purpose input output) provides various functionalities such as SPI, I²C, and serial universal asynchronous receiver-transmitter (UART) along with 3V3 and 5V power. The pins use a norm of 3V3 logic which can not accommodate 5V logic. In comparison to Raspberry Pi, other boards with almost similar capability which are available in market include Beagle Bone, Beagle Board, Panda Board, Odroid Board and many more. The Raspberry Pi has been chosen for this prototype since it encourages the purpose of learning, innovation and experimental studies with vibrant community support and introduces relatively cheaper development cost. The Central Processing Unit (CPU) within Raspberry Pi is the section that executes the given program [1]. It must be converted into a language the CPU can understand before a program can be run by a CPU. The ARM Central processing unit cannot recognize the instructions written for an Intel or an AMD CPU, ensuring that most of the technology off-shelf can not be run with a Raspberry Pi [1].

II. LITERATURE SURVEY

A microcontroller based hardware-friendly machine learning; Support Vector Machine (SVM) has been purposed for the application in automotive fields [12]. This paper exhibits a parallel cluster design for SVM-based article recognition, trying to demonstrate the points of interest, and execution benefits that come from a devoted equipment arrangement. The proposed equipment design gives parallel handling, asset sharing among the preparing units, and proficient memory the executives [7]. Besides, the measure of the exhibit is versatile to the equipment requests, and can likewise deal with an assortment of utilizations, for example, multiclass characterization issues. A model of the proposed engineering was actualized on a FPGA (field programmable gate array) stage and assessed utilizing three mainstream identification applications, showing continuous execution (40-122 fps for an assortment of utilizations) [13]. Monte-Carlo Simulation (MCS) is an incredible asset in taking care of unwavering quality issues. Not withstanding, the time has come expending use for the complex basic designing issues. Another generally utilized technique, First-Order Second Moment Method (FOSM) as a rule necessitates the qualities and subsidiaries of point of confinement state work. This paper introduces two kinds of unwavering performance investigation strategies based on Least Square Support Vector Machine (LS-SVM), such as LS-SVM-based Monte-Carlo Simulation (MCS) and LS-SVM-based First-Order Second Moment Method (FOSM). In the principal technique, LS-SVM is received to supplant the breaking point state capacity and upgrade the proficiency of figuring [14]. Getting an adequate number of tests or estimations is still tedious and exorbitant by and large. Along these lines, the issue of proficient gaining from a restricted preparing set turns out to be progressively imperative. Bolster vector machines (SVM) as an ongoing way to deal with characterization matter inside the system of factual learning hypothesis [18]. They execute classifiers of a movable adaptability, which is naturally enhanced on the preparation information for a decent speculation execution [15]. This article shown the implementation various machine learning algorithms like SVM, random forest and Naive Bayes and also emphasize the optimization of the deep learning algorithm to improve accuracy [19]. This paper proposed a technique to classify various machine learning algorithms and discussed features of performance evaluation by adapting different metrics. Multi domain applications could be designed using this technique [20].

III. CAN INTERFACING

CAN operate on carrier sense arbitration request priority protocol multiple access collision detection [5]. If a node concurrently tries to send a message to a CAN bus, it will also control the bus status as depicted in Fig. 2. It also sees if the bit sent by the node is in the bus. So this capability of CAN is called as carrier sense. If the bus is ideal all the nodes in a CAN bus are having equal opportunity to send a message on CAN, so all the nodes start initiation with the startup frame bit [6].

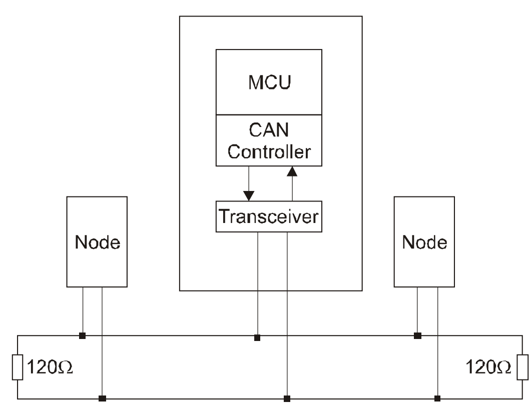

Fig. 2. Transmission of data via CAN bus.

This capability of CAN bus is called multiple access. If there is any collision in transmission concerned error flag will be set. This capability of CAN is called collision detection. The possibility is avoided by the long destructive bitwise arbitration process. Here the message priority will set by the identifier. If the value of identifier is less then message priority is more and if the identifier is more message priority is less.
Message priority will be less or more is decided by the programmer who is assigning the identifier value. So CAN is called arbitration based message priority protocol. Any time a node sends a message to CAN bus it will first see whether the node is free or not, before sending a message. If the bus is ideal all the nodes initiate transmission with the SOF bit but CAN bus allows to transmit only one message at a time so there should be an arbitration mechanism to decide which node has to send a message. The CAN interfacing with Ardunio and Raspberry Pi has been shown in Fig. 3 and 4.
CAN supports four different frame formats.
– Data Frame
– Remote Frame
– Error frame
– Overload frame
Data Frame and Remote frame are set by the user for transmission of message in the CAN bus. Remote Frame and Overload Frame are set internally by the CAN hardware using different flags to indicate different errors and overload conditions on CAN bus. Among four frames only two frames are discussed which are more important in CAN programming.

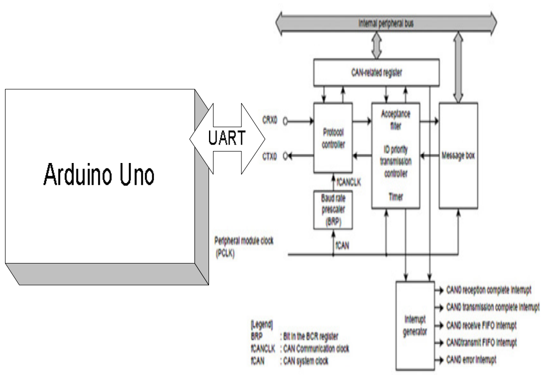

Fig. 3. Interfacing CAN bus with Arduino Uno.

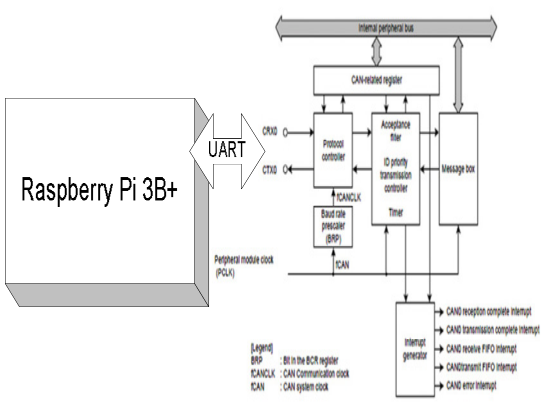

Fig. 4. Interfacing with Raspberry Pi.

Remote frame structure is same as DATA frame structure but absence of DATA field. In remote frame RTR bit is enabled and this frame is used to send a request to a transmitting node to send a requested message on a CAN bus from a particular node [2]. This RTR is using same identifier as a requested DATA frame on CAN bus. The receiver node will respond with same identifier as a requested node RTR frame identifier. Remaining all fields are same as DATA field. This 2 frame format plays a vital role in CAN programming. The article implement on real time data collected from sensor nodes deployed at agriculture field.

IV. IMPLEMENTATION ON HARDWARE PLATFORM

Raspberry Pi needs to interface with CAN bus.

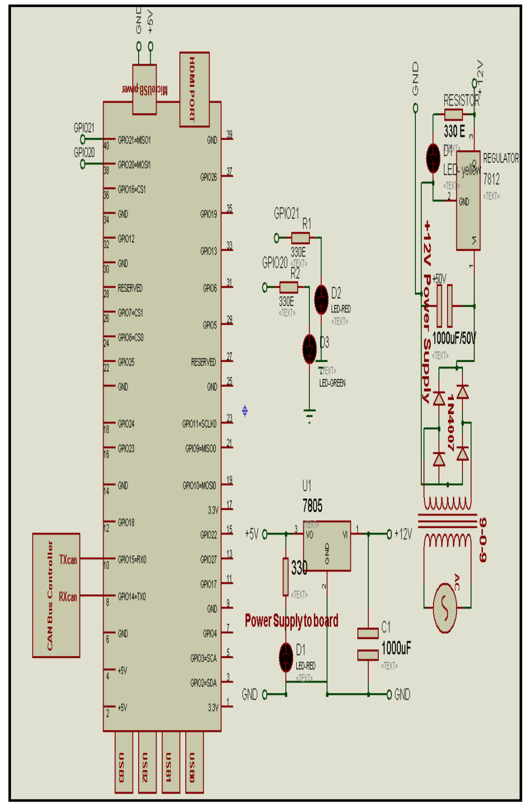

Fig. 5. Raspberry pi interfacing with CAN bus schematic circuit connection.

pi@raspberry:~ $ sudo apt-get update
pi@raspberry:~&sudo apt-get upgrade
Now we need to check the kernel version
pi@raspberry:~&uname –a
Installing can on raspberry board
pi@raspberry:~ $ sudo apt-get install can –utils
Setting the baud rate at 125000
pi@raspberry:~ $ sudoip link set can 0 up type can bitrate 125000
Sending some data via CAN bus:-
pi@raspberry:~$ can send can0 198359754568967
Now we can grab the data from CAN bus by putting the command
pi@raspberry:~$candump can0
The propagation time is defined as the double sum of the propagation time on the CAN bus and the delay of the comparator and the driver's output.
$T_{prop}=2*(t_{bus}+t_{comp}+t_{drive})$
T_{bus}- Ground trip time of physical bus
T_{comp}-input comparator delay
T_{drive}-output driver delay
The duration of propagation segment is 1Tq to 8Tq.

Normally bits are transmitted as electrical pulse so there is a problem of phase shift difference in each bit participating in arbitration process [9].

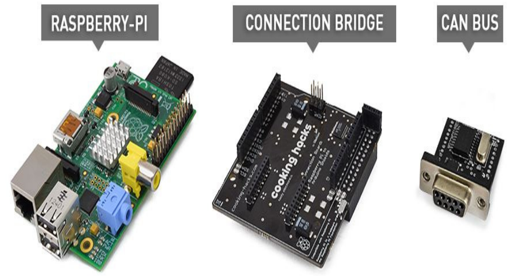

Fig. 6. Raspberry pi with CAN module.

These different phase shifts are adjusted on this phase segment 1 and 2 period. Fig. 5 depicts the Circuit connection of Raspberry Pi interfacing with CAN bus. These durations are modified by resynchronization. The duration of phase segment 1 is 1Tq to 8Tq. Phase 2 segment length is either the peak duration of Phase 1 segment or the Information Processing Time (IPT). Segment Phase 1 shall be extended or Segment 2 of Process shall be shortened when it is resynchronized. The lengthing or cutting of phase 1 and phase 2 buffer segments is defined by the upper bound of the 1Tq to 4Tq jump resynchronization width. Sample point is a point that reads and interprets the rate of the CAN bus as a value of that particular bit. Fig. 6 depicts the Raspberry Pi with Can module. Sample point is between the phase buffer section 1 and the phase buffer section 2 [3]. Sample point signifies the specific instant when the bit level is read and interpreted and transferred to CAN bus [10].

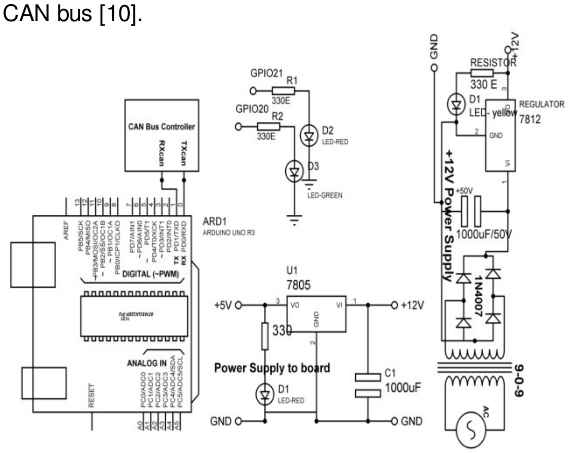

Fig. 7. Arduino Uno interfacing with CAN bus schematic circuit connection.

Number of bits transmitted on a CAN bus on an ideal transmitter per second without any synchronization. Nominal bit time is called the inverse of nominal bit rate. Nominal bit time=1/nominal bit rate. Total bit time is always lies in between 8Tq to 25Tq. Fig. 7 displays the schematic circuit connection of Ardunio Uno interfacing with CAN bus. CAN node will always initiated by configuration nodes. In Configuration mode, node neither transmits nor receives. It will clear error flag and interrupt flag remains unchanged [1, 2]. Time duration of each slice inside single bit is represented by a fundamental unit called Time quanta (Tq) which is derived from oscillator period there exist BRP. Synchronization segment is useful in synchronizing different bits participating on CAN bus as all the nodes edge of this section is supposed to lie inside.

V. SOFTWARE

Support vector machines technology is a surveillance-based system of training based on statistical theory and related algorithms for data analysis and pattern recognition [11]. By mapping the entrant vectors into the high-dimensional functional array of an extremely nonlinear fit, SVM produces a binary classifier, known as the optimum separating hyperplanes. From Embedded systems perspective SVM algorithm has been chosen [12].

A SVM algorithm:
Require: x and y loaded with training labelled data, $\alpha \Leftarrow 0$ or $\alpha \Leftarrow$partially trained SVM
$C \Leftarrow$some value (10 for example)
repeat
for all$\{x_i, y_i\}, \{x_j, y_j\}$do
    Optimize α_i and α_j
end for
Until no changes in α or other resource constraint criteria met
Ensure support vectors ($\alpha_i > 0$)

SVM is most powerful technique having high mathematical depth [8]. It is a guided learning algorithm based on the statistical method of learning. Kernel trick method is used in this learning, which projects data into higher dimensional space using kernel function K(x, z). It uses following mathematical decision function

$$D^{(Z)} = Sign \sum\nolimits_{i=1}^{n} (a_i y_i \cdot K(z, s_i) + b)$$

We perform the following normalization in order to build a hardware-friendly SVM [17]. Some of the standard vector operation has been followed. The machine learning paradigm is depicted in Fig. 8.

Linear: $K(x, z) = x \bullet z$
Polynomial: $K(x, z) = ((x \bullet z) + 1)^d$, $d > 0$

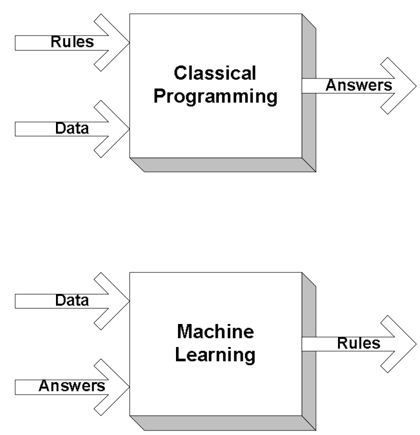

Fig. 8. The machine learning paradigm.

VI. RESULT AND DISCUSSION

There was an overview of the critical components of the SVM algorithm mapped between the software and hardware, showing how SVM can be accelerated with hardware calculations [12]. Fig 9 and 10 shows the applied algorithm and CAN initialization on the prototype. We have used TensorFlow tool for executing Artificial Intelligence algorithm. TensorFlow is open source software developed for IOTs, Automation and Embedded Systems etc. application by Google. The analysis shows the better target board to select to implement CAN protocol. The performance analysis helps us to decide controller according to our requirement. Fig. 11 shows the process of implementation of machine learning model.

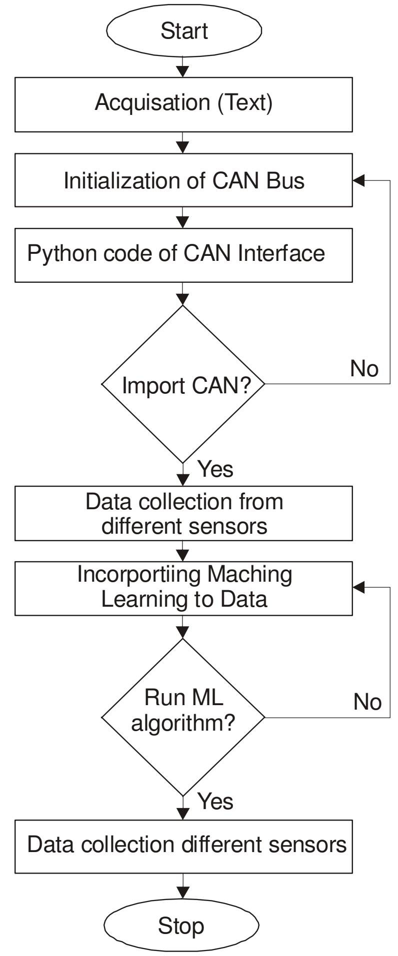

Fig. 9. Applied algorithm on Prototype.

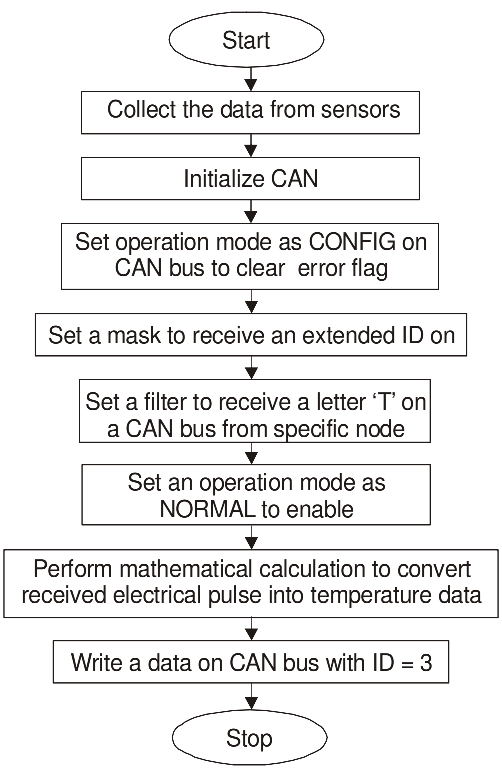

Fig. 10. CAN initialization on Prototype.

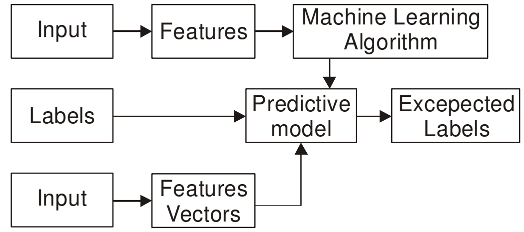

Fig. 11. Machine Learning Model implemented.

Table 1: Comparison of Evaluation parameters.

Evaluation parameters	Raspberry Pi	Arduino	Intel Edison
CPU cycles used	3.2×10^{18}	4.5×10^{18}	2.7×10^{18}
Context switch time	1239	1287	1342
Task clock cycle	4232	3954	3875
Cache hit time	1345	1356	1189
Overall performance	73%	65%	75%

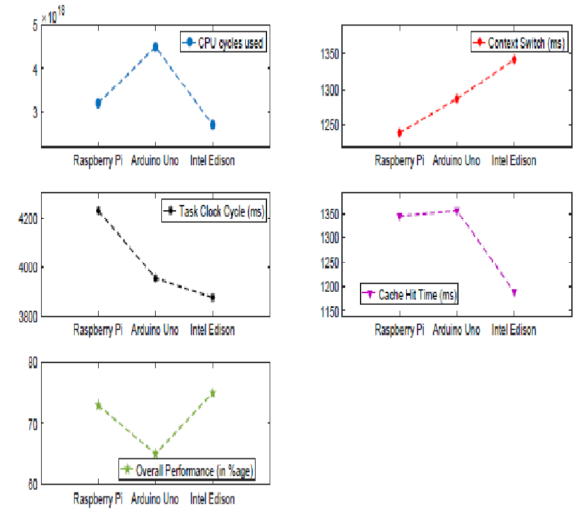

Fig. 12. Performance Analysis of evaluation parameters.

Table 2: Comparison of evaluation parameters.

Evaluation parameters	Raspberry Pi	Arduino	Intel Edison
Cumulative time in Sec	66.4	76.6	56.3
Function call time in sec	15.26	24.6	32.4
Overall Performance	54%	41%	64%

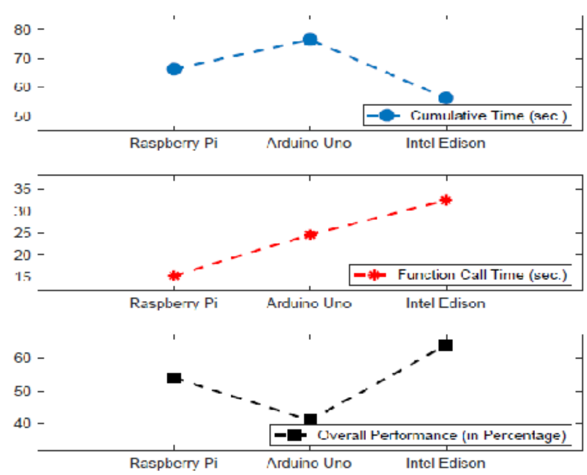

Fig. 13. Performance Analysis of evaluation parameters.

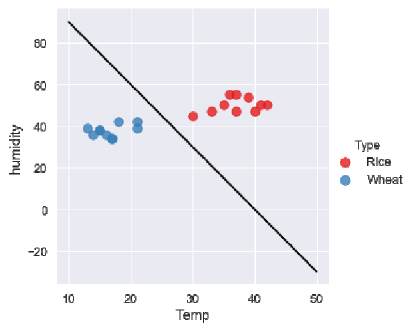

Fig. 14. Predictive outcome in agriculture using SVM on agricultural data.

VII. CONCLUSION

In this article, CAN protocol has been implemented on hardware platforms to collect the information's from various sensors and feeding in to Machine learning algorithm i.e SVM has been incorporated to predict the output. We have discussed three various hardware platforms for implementation i.e Raspberry Pi, Arduino and Intel Edison. The worst case time has been calculated using three algorithms. The performance has been compared and mentioned in Table 1, 2. Fig. 12 and 13. displays the graph of performance analysis of the said evaluation parameters. Fig. 14 shows the predictive outcomes analyzed using SVM. The approach employed in this article contains a variety of real-time applications. The areas of the applications are solving big data problem, weather prediction, implementation in smart city, military applications, trading and medicine to predict heart attack, cancer etc.

VIII. FUTURE SCOPE

CAN is highly robust and fault tolerant for errors. Raspberry pi with machine learning algorithm could be used for big data problem. The prototype can be implemented in Smart cities, smart home, computational biology, energy production etc.

Conflict of Interest. No potential conflict of interest.

REFERENCES

[1]. Shin, C. M., Han, T. M., Ham, H. S., & Lee, W. J. (2010). Method for transmitting/receiving data frame in can protocol. *U.S. Patent Application No.* 12/543,876.

[2]. Woo, S., Jo, H. J., & Lee, D. H. (2015). A practical wireless attack on the connected car and security protocol for in-vehicle CAN. *IEEE Transactions on intelligent transportation systems*, *16*(2), 993-1006.

[3]. Bosch, R. (1991). CAN Specification 2.0. Robert Bosch GmbH, Postfach, Stuttgart, Germany.

[4]. Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., & Savage, S. (2010). Experimental security analysis of a modern automobile. In *2010 IEEE Symposium on Security and Privacy* (pp. 447-462). IEEE.

[5]. Nahas, M. (2011). Applying Eight-to-Eleven Modulation to reduce message-length variations in distributed embedded systems using the Controller Area Network (CAN) protocol. *Canadian Journal on Electrical and Electronics Engineering*, *2*(7), 282-293.

[6]. Yu, J., Qi, Y., Wang, G., & Gu, X. (2012). A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution. *AEU-International Journal of Electronics and Communications*, *66*(1), 54-61.

[7]. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: a system for large-scale machine learning. In *OSDI* , *16*, 265-283.

[8]. Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

[9]. Matthews, D. G., Alexander, G., Van Der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A & Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *The Journal of Machine Learning Research*, *18*(1), 1299-1304

[10]. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. *1*). Cambridge: MIT press.

[11]. Barr, T. W., Smith, R., & Rixner, S. (2012). Design and implementation of an embedded Python run-time system. In *Presented as part of the 2012 {USENIX} Annual Technical Conference ({USENIX}{ATC} 12)* (pp. 297-308).

[12]. Anguita, D., Ghio, A., Pischiutta, S., & Ridella, S. (2007). A hardware-friendly support vector machine for embedded automotive applications. In *2007 international joint conference on neural* networks (pp. 1360-1364). IEEE.

[13]. Kyrkou, C., & Theocharides, T. (2012). A parallel hardware architecture for real-time object detection with support vector machines. *IEEE Transactions on Computers, 61*(6), 831-842

[14]. Zhu, C., & Zhao, H. (2016). Least square support vector machine for structural reliability analysis. *International Journal of Computer Applications in Technology, 53*(1), 51-61.

[15]. Belousov, A. I., Verzakov, S. A., & Von Frese, J. (2002). A flexible classification approach with optimal generalisation performance: support vector machines. *Chemometrics and intelligent laboratory systems, 64*(1), 15-25.

[16]. Yin, Z., & Hou, J. (2016). Recent advances on SVM based fault diagnosis and process monitoring in complicated industrial processes. *Neurocomputing, 174*, 643-650.

[17]. Pantazi, X. E., Moshou, D., Alexandridis, T., Whetton, R. L., & Mouazen, A. M. (2016). Wheat yield prediction using machine learning and advanced sensing techniques. *Computers and Electronics in Agriculture, 121*, 57-65.

[18]. Wen, Z., Shi, J., Li, Q., He, B., & Chen, J. (2018). Thunder SVM: A fast SVM library on GPUs and CPUs. The *Journal of Machine Learning Research, 19*(1), 797-801.

[19]. Nair, R., & Bhagat, A. (2019). Genes Expression Classification using Improved Deep Learning Method. *International Journal on Emerging Technologies, 10*(3), 64-68.

[20]. Priya, N., & Radhika, C. (2019). A Survey on Predicting Autism Spectrum Disorder using Machine Learning techniques. *International Journal on Emerging Technologies, 10*(3), 422-427.

How to cite this article: Swain, Mahendra, Singh, Rajesh, Thakur, Amit Kumar and Gehlot, Anita (2020). A Machine Learning Approach of Data Mining in Agriculture 4.0. *International Journal on Emerging Technologies*, *11*(1): 257–262.