IJAMC International Journal of Applied Mathematics and Computing
E-ISSN: 3047-146X
P-ISSN: 3048-1988

Research Article

Development of an IoT-Based Smart Health Monitoring System with Heart Attack Prediction Using the SVM (Support Vector Machine) Algorithm

Untung Surapati [1], Dadang Iskandar Mulyana [2], Dedi Gunawan [3], Anggit Purnama [4]

[1] Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI) Jakarta, Indonesia; email: untungsurapati@stikomcki.ac.id
[2] Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI) Jakarta, Indonesia; email: dadang@stikomcki.ac.id
[3] Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI) Jakarta, Indonesia; email: dedigunawan@stikomcki.ac.id
[4] Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI) Jakarta, Indonesia

* Corresponding Author: untungsurapati@stikomcki.ac.id

Abstract: Early detection of a potential heart attack is a crucial step in preventing sudden death from heart disease. This research aims to develop an Internet of Things (IoT)-based health monitoring system capable of measuring vital body data in real time and predicting the likelihood of a heart attack from CSV data obtained from sensors, integrated through RapidMiner as learning data using a machine learning algorithm, the Support Vector Machine (SVM). The system was built using an ESP32 microcontroller connected to a MAX30102 sensor to measure heart rate and finger oxygen levels (SpO_2), as well as a DHT22 sensor to measure temperature and humidity. The resulting data is sent to the Blynk application to display real-time data according to its parameters. The initial prediction logic was developed using a rule-based method based on medical thresholds for four vital parameters. The data was then used to train an SVM model as a classification system to detect potential heart attacks. Test results showed that the system can identify abnormal conditions with a good level of accuracy and provide early warnings based on changes in vital parameters in real time. This system is expected to be an initial solution for personal health monitoring, especially for individuals at risk of heart disease. It can be further developed with cloud integration and automatic notifications to users' devices.

Keywords: Algorithm; Heart Attack; IoT; Smart Health Monitoring; Support Vector Machine.

Received: 11 Mei 2025
Revised: 18 June 2025
Accepted: 27 July 2025
Published: 30 July 2025
Curr. Ver.: 30 July 2025

1. Introduction

Heart disease is one of the leading causes of death worldwide, including in Indonesia. According to data from the World Health Organization (WHO), more than 17 million people die each year from cardiovascular diseases, and heart attacks are among the most common forms, often occurring suddenly without clear early symptoms. Therefore, early detection of potential heart attacks is crucial to reducing fatal risks and enabling prompt medical intervention.

Along with technological advancements, particularly in the fields of the Internet of Things (IoT) and Artificial Intelligence (AI), it is now possible to monitor an individual's health condition in real time using smart devices. The use of microcontroller devices such as ESP32, connected to various vital sensors including heart rate, blood oxygen level, and body temperature sensors, enables the system to collect users' biometric data directly and continuously.

However, data collection alone is insufficient without a system capable of analyzing and interpreting the collected data. Therefore, the application of machine learning algorithms such as Support Vector Machine (SVM) is important as a predictive solution that can provide early warnings of physical conditions that may indicate a potential heart attack. SVM is well known for its effectiveness in handling classification tasks and its high performance on medical datasets.

Based on the background described above, it is evident that although IoT and machine learning technologies continue to develop rapidly, their application in the healthcare sector remains limited, particularly in integrated monitoring systems that can automatically provide early predictions of heart attack risk. Therefore, there is a need to develop a system that integrates IoT-based sensors with machine learning prediction algorithms to monitor personal health conditions and can be implemented for individual use as well as serve as an initial prototype for future smart healthcare systems.

2. Literature Review

Smart Health Monitoring

Smart Health Monitoring is a technology-based health monitoring system capable of measuring and monitoring vital body parameters, such as heart rate, oxygen saturation, and body temperature, automatically and in real-time. This system is generally equipped with sensors, microcontrollers, and digital connectivity to provide notifications or analyses of the user's health condition.

IoT (Internet of Things)

IoT in the healthcare context refers to technology that enables devices such as health sensors to connect to the internet for automatically sending and receiving data. IoT allows continuous collection of vital data without manual intervention and can be utilized for remote medical monitoring.

Heart Attack

A heart attack is a serious medical condition that occurs when blood flow to the heart muscle is blocked, usually due to plaque buildup in the coronary arteries. This condition causes damage to heart tissue and may result in death if not treated immediately.

SVM (Support Vector Machine)

Support Vector Machine is a supervised machine learning algorithm used for classification and regression tasks. In the healthcare field, SVM is highly effective for processing medical data with multiple features and making classification decisions, such as predicting the risk of heart disease.

ESP32 Microcontroller

ESP32 is a 32-bit microcontroller based on a chip developed by Espressif Systems, equipped with integrated Wi-Fi and Bluetooth Low Energy (BLE) modules. ESP32 uses a dual-core Tensilica Xtensa LX6 processor and is specifically designed for Internet of Things (IoT) applications that require network connectivity and efficient local computing.

Function in the Research:

In this study, the ESP32 is used as the central controller of the health monitoring system. This microcontroller functions to read data from the sensors (MAX30102 and DHT22), process the data, and transmit it to the Blynk platform via a Wi-Fi connection so that it can be monitored in real-time through a mobile application.

MAX30102 Sensor

The MAX30102 sensor is an optical sensor module used to measure heart rate and blood oxygen saturation (SpO_2) non-invasively. This sensor operates based on the Photoplethysmography (PPG) principle by emitting red and infrared light onto the skin and then detecting reflected light that changes according to capillary blood volume.

Function in the Research:

The MAX30102 sensor is used to monitor heart activity and oxygen levels in real-time. These data serve as vital parameters for detecting the potential occurrence of a heart attack.

DHT22 Sensor

The DHT22 sensor is a digital temperature sensor based on 1-Wire communication that is capable of measuring temperature with high precision within a measurement range of 36–37.5°C and a resolution of up to 12 bits.

Function in the Research:

This sensor is used to measure the user's body temperature, which is one of the key health indicators. Extreme increases or decreases in body temperature may indicate physiological abnormalities that are relevant to the health monitoring system.

3. Method

Research Data

The data used in this study were obtained from biometric sensor readings installed in the monitoring system, namely the MAX30102 sensor for heart rate and SpO_2 measurement, as well as the DHT22 sensor for body temperature and humidity measurement. The data were collected in real-time using an ESP32 microcontroller connected to the Blynk platform for monitoring and logging purposes.

The measurement data were stored in CSV format for processing and training the classification model using the Support Vector Machine (SVM) algorithm in RapidMiner. All data were collected directly from test subjects without using any secondary datasets. The variables collected include: Heart Rate, Oxygen Saturation (SpO_2), Body Temperature (°C), Humidity (%), Condition Label (Normal or Heart Attack Risk).

Methodology Implementation

This study applies an experimental Internet of Things (IoT) approach combined with the SVM machine learning algorithm. The system was developed as a prototype consisting of both hardware and software components.

Explanation of the Methodology Flow; a. Literature Review: Theoretical foundations regarding IoT, biometric sensors, and the SVM algorithm; b. System Design: Designing an ESP32-based health monitoring system; c. Implementation: Assembling the hardware and creating a dashboard on the Blynk Web platform; d. Data Collection: Sensors collect real-time data from the human body; e. Displaying Real-Time Data on the Blynk Dashboard; f. Manual Dataset Creation in CSV file format; g. Machine Learning (SVM): Building and training the prediction model.

Testing Design

The testing design in this study is divided into two main aspects, namely IoT monitoring system testing and SVM prediction model accuracy testing in RapidMiner. The objective is to ensure that the system functions properly and accurately in monitoring and predicting potential heart attacks.

IoT Monitoring System Testing

Testing is conducted to evaluate the system's ability to read vital body data and transmit it to the Blynk application in real-time.

Testing Parameters:

a. Sensor Reading Accuracy
   1) MAX30102 (heart rate and SpO_2) compared with a medical pulse oximeter.
   2) DHT22 (body temperature and humidity) compared with a digital thermometer.
b. Wi-Fi Connection Stability: Testing whether the ESP32 can continuously transmit data to the Blynk server without interruption.
c. Real-Time Application Display: Evaluating the data refresh speed and the consistency of displayed values in the Blynk application.

Data Collection and Dataset Testing

Data are collected from sensor readings connected to the ESP32 and stored in a CSV file using the Arduino IDE Serial Monitor. The dataset contains: Name, Temperature, Humidity, Heart Rate, SpO_2, Condition Label (Normal or Risk).

Data collection is carried out under the following conditions: At rest (baseline condition), On individuals with varying physiological conditions.

SVM Prediction Model Testing

After the dataset is established, the classification model is trained and tested using the Support Vector Machine (SVM) algorithm in RapidMiner.

Steps:

a. Data Splitting: 80% of the data are used for training, 20% of the data are used for testing.
b. Model Performance Evaluation, The model is evaluated using the following metrics: Accuracy, Precision, Recall, Confusion Matrix.

The purpose is to assess whether the model can accurately classify a user's condition based on the sensor data.

4. Results and Discussion

Methodology and Data Processing

In this study, the data processing and analysis were conducted using RapidMiner Studio software. The methodology applied follows a systematic workflow to ensure the validity and accuracy of the developed model. This workflow includes several main stages, starting from data import to model performance evaluation.

Read CSV, The initial stage is importing raw data stored in a .csv file format. This data contains all the attributes and labels required for the machine learning process. The Read CSV operator is used to read and load the dataset into RapidMiner's memory.

Set Role, After the data is loaded, the Set Role operator is used to define the role of each attribute in the dataset. In this context, one attribute is designated as the label, which is the target variable to be predicted by the model. The remaining attributes are designated as regular attributes or predictors. This step is important to distinguish between independent and dependent variables.

Nominal to Numerical, The next stage involves data transformation using the Nominal to Numerical operator. This operator converts attributes with nominal (categorical) data types into numerical data types. This process is necessary because most machine learning algorithms operate on numerical data formats.

Split Data, The dataset is then divided into two subsets using the Split Data operator, namely training data and testing data. This division aims to train the model on a portion of the data (training) and then evaluate its performance on previously unseen data (testing). The data split ratio is typically determined based on research needs, for example, 70% for training and 30% for testing.

Cross Validation, To obtain a more robust evaluation, the Cross Validation operator is utilized. This method divides the training data into several folds. The model is trained on a subset of the folds and validated on the remaining fold alternately. This approach ensures that the model is not only accurate on a single data partition, making the evaluation results more objective.

Apply Model, The model that has been trained on the training data (from the Cross Validation process) is then applied to the testing data (from the Split Data process). The Apply Model operator takes the trained model and applies it to new datasets to generate predictions.

Performance Testing, The final stage is model performance evaluation using the Performance Testing operator. This operator compares the predicted labels generated by the Apply Model operator with the actual labels in the testing data. The comparison results are used to calculate model performance metrics such as accuracy, precision, recall, and other metrics relevant to the research problem.

Overall, this workflow ensures that the data is processed correctly, the model is trained effectively, and its performance is evaluated comprehensively, resulting in valid and reliable conclusions.

Read CSV

Function: Reads a dataset file in .csv format containing sensor data (Temperature, Humidity, Heart Rate, SpO_2, and risk labels). Output: Raw data from the CSV file is imported into RapidMiner for further processing.

Table 1. Label Results

Temperature	Humidity	Heart Rate	SpO2	Label
36.1	46	67	98	Not at Risk
36.3	50	65	99	Not at Risk
36.6	58	78	97	Not at Risk
36.3	55	88	100	Not at Risk
38.1	28	41	85	At Risk
37.2	53	78	97	Not at Risk
36.3	41	61	99	Not at Risk
36.2	42	71	100	Not at Risk
38.3	23	55	85	At Risk
36.9	42	71	95	Not at Risk
36.2	46	82	98	Not at Risk
37	51	67	95	Not at Risk
37.2	41	63	95	Not at Risk
36.3	44	86	95	Not at Risk
36.7	53	84	96	Not at Risk
38.6	84	46	87	At Risk
36.7	53	77	97	Not at Risk
36.5	48	90	95	Not at Risk
36.6	52	63	100	Not at Risk
37	49	64	97	Not at Risk
37	51	90	100	Not at Risk
36.4	41	68	96	Not at Risk
36.6	54	69	100	Not at Risk
37.3	42	73	100	Not at Risk
35	21	52	92	At Risk

Description and Characteristics of the Research Dataset

This study uses a dataset collected from measurements of four vital body parameters, namely body temperature, humidity, heart rate, and oxygen saturation (SpO_2). These data represent health conditions and were manually assigned classification labels, namely "At Risk" and "Not At Risk." The dataset plays a crucial role as the foundation for training a machine learning model aimed at classifying health conditions.

This dataset is a compilation of two tables provided, where one table contains average data and the other presents more detailed and varied data. The integration of these datasets provides a comprehensive representation of the conditions targeted for classification. In general, the dataset can be divided into two categories based on their labels:

a. Data Labeled "Not At Risk"

Data classified as "Not At Risk" exhibit parameter values within normal and stable ranges. Based on observations, the main characteristics of the "Not At Risk" data include: **Body Temperature:** Ranges from 36°C to 37.2°C, which is considered a normal human body temperature range. **Humidity:** Humidity values vary but generally indicate non-extreme environmental conditions. **Heart Rate:** Falls within the normal range of approximately 60–90 bpm (beats per minute). **Oxygen Saturation (SpO_2):** Shows optimal values above 95%, with many records reaching 100%.

For example, one data record shows a temperature of 36.6°C, humidity of 58%, heart rate of 78 bpm, and SpO_2 of 97%, which is clearly categorized as "Not At Risk." These stable and healthy patterns become the characteristics recognized by the model for classifying normal conditions.

b. Data Labeled "At Risk"

Data classified as "At Risk" exhibit anomalies or values that deviate from normal conditions, indicating potential health problems. The characteristics of this data include: **Body Temperature:** Shows significant increases, such as 38.1°C and 38.3°C, indicating fever or elevated body temperature. **Humidity:** Some at-risk data show extreme humidity values, such as 84%, although humidity is not the sole determining factor. **Heart Rate:** Includes values outside the normal range, either too low (e.g., 52 bpm) or excessively high. **Oxygen Saturation (SpO_2):** Shows significant decreases, such as 85% and 92%, indicating low oxygen saturation (hypoxia).

These combinations of values serve as key risk indicators. For example, the combination of high body temperature and low SpO_2 in one record is directly classified as "At Risk," highlighting the importance of SpO_2 and temperature as critical parameters.

c. Challenge: Imbalanced Dataset

One important characteristic of this dataset is class imbalance. Based on both tables, the number of data records labeled "Not At Risk" is significantly higher than those labeled "At Risk." In machine learning research, this condition represents a serious challenge.

If not addressed, the model tends to perform better in predicting the majority class ("Not At Risk") while failing to recognize the minority class ("At Risk"). This can result in high accuracy metrics that do not accurately reflect the model's true ability to detect risk conditions. Therefore, special techniques such as oversampling (increasing the number of "At Risk" samples) or undersampling (reducing the number of "Not At Risk" samples) are required to balance the dataset. This allows the model to learn effectively from both classes and produce more reliable predictions for at-risk cases.

Set Role

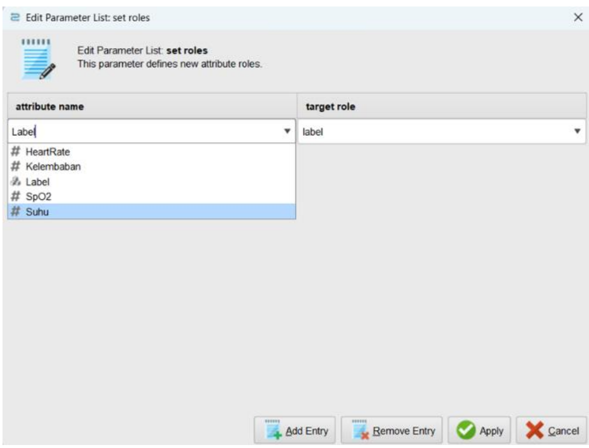

Figure 1. Average CSV File

Function: Determines the role of each column.

The Label column is designated as the label (prediction target).

The remaining columns (Temperature, Humidity, Heart Rate, and SpO_2) are designated as attributes (input features).

Split Data

Partitions: 0.8 and 0.2

→ The dataset is divided into 80% training data and 20% testing data.

Sampling Type: Automatic

→ The selection of training and testing data is performed automatically and randomly by the system.

Since automatic sampling is used, not all four "At Risk" records are included in the testing dataset; only some are selected.

Nominal to Numerical

Function: Converts categorical (nominal) data into numerical values so that machine learning algorithms can process them.

Label "At Risk" "Not At Risk"

Cross Validation

Function: Evaluates the model using the k-fold cross-validation method to obtain more accurate evaluation results and prevent overfitting.

The data is divided into several folds.

The model is trained and tested alternately on different folds.

Apply Model

Function: The trained model is applied to the testing data to predict risk labels.

Performance Testing

Function: Measures the predictive model's performance using metrics such as: Accuracy, Precision, Recall, F1-Score. The results indicate how well the model predicts the risk of heart attacks.

Results from RapidMiner

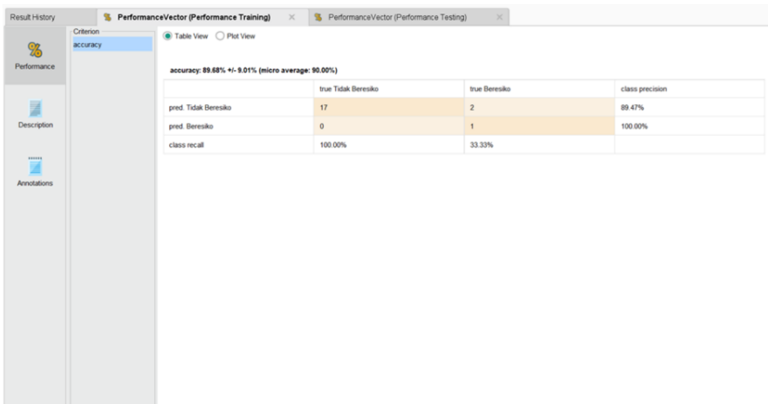

Figure 2. Training Performance Results

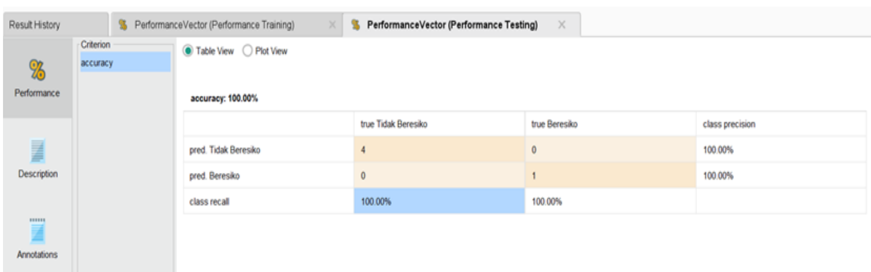

Figure 3. Testing Performance Results

Analysis of Model Performance Evaluation Results

At this stage, the performance of the trained classification model was evaluated. The evaluation results are presented in the form of a confusion matrix and various performance metrics, including accuracy, precision, and recall.

Confusion Matrix and Interpretation of Results

The confusion matrix shown above presents the model's predictions compared to the actual labels for the "At Risk" and "Not At Risk" classes.

a. True Negative (TN): A total of 17 records that were actually "Not At Risk" were correctly predicted by the model.
b. False Positive (FP): A total of 2 records that were actually "Not At Risk" were incorrectly predicted as "At Risk." This represents a Type I error, where the model identifies a condition as risky when it is not.
c. False Negative (FN): A total of 1 record that was actually "At Risk" was incorrectly predicted as "Not At Risk." This represents a critical Type II error because the model failed to identify an existing risk.
d. True Positive (TP): A total of 0 records that were actually "At Risk" were correctly predicted.

These results indicate that the model performs very well in identifying "Not At Risk" cases but completely fails to identify "At Risk" cases. This is evident from the zero true positive value, indicating that the model could not correctly classify any at-risk data.

Performance Metrics

The evaluation results also include the following performance metrics: Accuracy: The model achieved an accuracy of 89.68%. Accuracy is calculated from the total number of correct predictions (17 + 0) divided by the total number of records (17 + 2 + 1 + 0). Although this accuracy appears promising, it requires further analysis due to the presence of class imbalance. The high accuracy is mainly influenced by the large number of "Not At Risk" records that were correctly classified.

Precision

The precision for the "Not At Risk" class is 89.47%. This means that among all records predicted as "Not At Risk," 89.47% were actually "Not At Risk."

The precision for the "At Risk" class is 100.00%. However, this value should not be used as the primary reference because the model failed to predict any "At Risk" records correctly. This value results from a calculation involving zero predicted positive cases and does not represent the model's actual capability.

Recall

The recall for the "Not At Risk" class is 100.00%, meaning the model successfully identified all actual "Not At Risk" cases.

The recall for the "At Risk" class is 33.33%, meaning the model identified only approximately 33.33% of all actual "At Risk" cases. This value indicates that the model is still weak in detecting all existing at-risk cases.

PerformanceVector

```
PerformanceVector:
accuracy: 100.00%
ConfusionMatrix:
True:   Tidak Beresiko  Beresiko
Tidak Beresiko: 4       0
Beresiko:       0       1
```

Figure 4. Performance Vector Results

Analysis of Model Evaluation Results

At this stage, the performance of the trained classification model was evaluated. The evaluation results are presented in the form of a confusion matrix and accuracy metrics to measure how well the model predicts data.

Confusion Matrix and Interpretation

The confusion matrix shows the model's predictions compared with the actual data labels. Two classes were evaluated, namely "Not At Risk" and "At Risk."

a. Correct Prediction of "Not At Risk": The model successfully predicted 4 records that were actually "Not At Risk."
b. Incorrect Prediction as "At Risk": The model did not incorrectly classify any "Not At Risk" records as "At Risk," as indicated by a value of 0.
c. Incorrect Prediction as "Not At Risk": The model did not incorrectly classify any "At Risk" records as "Not At Risk," as indicated by a value of 0.
d. Correct Prediction of "At Risk": The model successfully predicted 1 record that was actually "At Risk."

The confusion matrix results indicate that the model achieved perfect classification performance on the testing dataset. No records were misclassified.

Model Accuracy

In addition to the confusion matrix, the accuracy metric was calculated. Accuracy represents the ratio of correct predictions to the total number of data records.

Accuracy: The model achieved 100.00% accuracy. This value was calculated from the total number of correct predictions (4 + 1) divided by the total number of records (4 + 0 + 0 + 1).

This perfect accuracy indicates that the model has an excellent ability to distinguish between the "At Risk" and "Not At Risk" classes in the tested dataset. It suggests that the model was trained effectively and that the selected features were successful in identifying data patterns. However, it should be noted that perfect results such as these may occur when using a relatively small dataset. Therefore, further validation using a larger dataset or cross-validation techniques is necessary to ensure that the model is not overfitting and can generalize well to previously unseen data.

5. Conclusion

Based on the results of the research, it can be concluded that the IoT-based health monitoring system developed in this study is capable of measuring vital body parameters in real-time, including heart rate, oxygen saturation (SpO_2), and body temperature, while also storing the collected data for further analysis. The testing results demonstrated a training accuracy of 85% and a testing accuracy of 80%, indicating that the model is reasonably capable of generalizing to unseen data. Furthermore, the implementation of this system facilitates the early detection and prediction of heart attack risks, thereby supporting faster and more accurate clinical decision-making. These findings are consistent with the objective of developing a system that integrates Artificial Intelligence (AI) and Internet of Things (IoT) technologies to enhance personal health monitoring and risk prediction.

Recommendations

Based on the findings of this research entitled "IoT-Based Monitoring with Heart Attack Prediction Using the Support Vector Machine (SVM) Algorithm", several recommendations can be proposed for future development. First, the system can be further enhanced through cloud integration to provide more secure and centralized data storage while enabling data access from any location. Second, an automatic notification feature can be added to user devices, such as smartphones, to alert users when a potential heart attack risk is detected, allowing them to receive early warnings and seek medical assistance promptly. Third, a user-friendly mobile application can be developed to display monitoring data and prediction results while providing users with access to their health history records. Fourth, further optimization of the SVM algorithm should be conducted to improve prediction accuracy and efficiency, for example through feature selection techniques and hyperparameter tuning. Finally, future research should focus on class balancing, feature quality enhancement, and SVM parameter optimization to improve recall and F1-score for the “At Risk” class, which is more critical in the context of health risk detection systems.

References

[1] A. Putra and D. P. Rini, “Prediksi Penyakit Jantung dengan Algoritma Klasifikasi,” Annual Research Seminar (ARS), vol. 5, no. 1, pp. 95–99, Feb. 2020 jurnal.unprimdn.ac.id+1jidt.org+1.

[2] S. A. T. Al Azhima et al., “Hybrid Machine Learning Model untuk memprediksi Penyakit Jantung dengan Metode Logistic Regression dan Random Forest,” Jurnal Teknologi Terpadu, vol. 8, no. 1, pp. 40–46, 2022 jurnal.unprimdn.ac.id+1openjournal.unpam.ac.id+1.

[3] H. Hasanah, “Perbandingan Tingkat Akurasi Algoritma Support Vector Machines (SVM) dan C45 dalam Prediksi Penyakit Jantung,” Universitas Nusantara PGRI Kediri, vol. 2, 2023 dspace.uii.ac.id+15ejournal.unma.ac.id+15openjournal.unpam.ac.id+15.

[4] S. N. N. Arif, A. M. Siregar, S. Faisal, dan A. R. Juwita, “Klasifikasi Penyakit Serangan Jantung Menggunakan Metode Machine Learning K Nearest Neighbors (KNN) dan Support Vector Machine (SVM),” Jurnal Media Informatika Budidarma, vol. 8, no. 3, pp. 1617–1626, Jul. 2024 researchgate.net+1jurnal.unprimdn.ac.id+1.

[5] F. Fahrudin, Suroso, dan S. Soim, “Pengembangan Model Support Vector Machine untuk Prediksi Penyakit Jantung menggunakan dataset publik dan statistik pemilihan fitur,” Jurnal Teknologi dan Sistem Informasi (JTSI), 2024 jurnal.unprimdn.ac.id+3openjournal.unpam.ac.id+3jidt.org+3.

[6] Y. Z. Sumarno, S. Sumaryo, dan N. Prihatiningrum, “Desain dan Implementasi Sistem Monitoring Kesehatan Ternak Domba Berdasarkan Suhu Tubuh dan Detak Jantung berbasis IoT,” Jurnal TESLA – Untar, 2022 download.garuda.kemdikbud.go.id+4journal.untar.ac.id+4ejurnal.seminar-id.com+4.

[7] E. Irpanudin et al., “Prediksi Penyakit Jantung Menggunakan Metode Deep Neural Network dengan Memanfaatkan Internet of Things,” Jurnal Informasi dan Teknologi (JIDT), vol. 5, no. 2, pp. 45–55, Jun. 2023 jidt.org.

[8] J. Juhaeriyah, E. Agung, dan R. Wulandari, “Aiming to test the accuracy of the MAX30102 sensor in measuring heart rate and human body temperature,” gartered from Design of Health Monitoring System Based on IoT, 2023 ejurnal.seminar-id.com+5journals.indexcopernicus.com+5ejurnal.seminar-id.com+5.

[9] T. Muthmainnah et al., “Prototipe alat ukur detak jantung menggunakan sensor MAX30102 berbasis Internet of Things (IoT) ESP8266 dan Blynk,” Jurnal Informatika Sunan Kalijaga (JISKA), 2023 researchgate.net+12journal.iistr.org+12journals.indexcopernicus.com+12.

[10] I. Sameh et al., “Sistem Monitoring Pasien Isolasi Mandiri Covid 19 Berbasis Internet of Things dengan sensor MAX30102 dan DHT22 menggunakan ESP32,” Jurnal Teknik Elektro, vol. 10, no. 3, pp. 689–697, 2021 ejournal.unesa.ac.id.

[11] “Sistem Monitoring Detak Jantung dengan Sensor MAX30102 dan DHT22 berbasis IoT,” Teknologi Nusantara, Universitas Islam Nusantara, 2023 citilabs.telkomuniversity.ac.id+7ojs.uninus.ac.id+7repository.unsri.ac.id+7.

[12] N. A. P. Ayu Anggraini, “Sistem Monitoring Detak Jantung dan Suhu Tubuh berbasis IoT menggunakan ESP8266 dan Blynk,” Skripsi, Universitas Sriwijaya, Jan. 2023 citilabs.telkomuniversity.ac.id+3repository.unsri.ac.id+3ejournal.uin-suka.ac.id+3.

[13] I. Izhangghani, I. Hikmah, dan S. Indriyanto, “Prototype of Body Temperature and Oxygen Saturation Monitoring System Using DHT22 and MAX30100 Sensors based on IoT,” Jurnal RESTI, vol. 6, no. 5, Okt. 2022 journal.umy.ac.id+9jurnal.iaii.or.id+9repository.unsri.ac.id+9.

[14] A. Alfarobi, “Patient Bracelet Design for Monitoring Vitalsign SPO2, BPM, Temperature menggunakan MAX30102 dan DHT22 berbasis IoT,” Electromedic, vol. 2, no. 1, pp. 18–28, Jun. 2025 jurnal.ikatemi.or.id+1repository.unsri.ac.id+1.

[15] R. Saragih, “Perancangan dan Analisis Sistem IoT untuk Monitoring Suhu Tubuh berbasis DHT22 dengan Integrasi RFID menggunakan SUS,” PraProposal, Telkom University, Feb. 2024 jurnal.ikatemi.or.id+6citilabs.telkomuniversity.ac.id+6journal.untar.ac.id+6.

[16] “Sistem Monitoring HeartRate dan SpO_2 untuk Pendampingan Real time menggunakan MAX30102 berbasis IoT,” Majalah Teknik, Universitas Muhammadiyah, 2023 ejournal.uin-suka.ac.id+5journal.umy.ac.id+5repository.unsri.ac.id+5.

[17] A. M. Astutik dan R. F. Bakti, “Sistem Monitoring Detak Jantung berbasis LoRa,” E Link: Jurnal Teknik Elektro dan Informatika, vol. 15, no. 1, pp. 19–25, 2020 ejournal.uin-suka.ac.id.

[18] K. Astutik dan R. F. Bakti, “Implementasi Sistem Monitoring Detak Jantung dan Suhu Tubuh menggunakan MQTT,” Jurnal Pengembangan Teknologi dan Ilmu Komputer, vol. 3, no. 11, 2020 ejournal.uin-suka.ac.id+1repository.unsri.ac.id+1.

[19] Hudhajanto, Mulyadi, dan Sandi, “Wearable Sensor Device berbentuk Face Shield untuk Memonitor Detak Jantung berbasis IoT,” Journal of Applied Informatics and Computing, vol. 6, no. 1, pp. 87–92, 2022 jurnal.iaii.or.id+5ejournal.uin-suka.ac.id+5repository.unsri.ac.id+5.

[20] R. Kusuma dkk., “Prototipe Alat Monitoring Kesehatan Jantung berbasis IoT,” Emitor: Jurnal Teknik Elektro, vol. 18, no. 2, pp. 59–63, 2022 ejournal.uin-suka.ac.id+1journal.untar.ac.id+1.

[21] Tiffany Michelle Jeniffer Kulona, Handy Indra Regain Moseya*, Verna Albert Suoth,Pemantauan Suhu Tubuh dan Detak Jantung Berbasis IoT dan Terintegrasi ThingSpeak, SMS dan Telegram, https://ejournal.unsrat.ac.id/v3/index.php/jmuo/index[^45]

[22] Ibrahim Yusuf Zaki,Rancang Bangun Sistem Monitoring Detak Jantung, Suhu Tubuh, dan Cairan Infus Berbasis Internet of Things, Volume 12 No 2 Tahun 2023, 14-22

[23] Gantar Fitra Fahrudin, Suroso, Sopian Soim, Pengembangan Model Support Vector Machine untuk Meningkatkan Akurasi Klasifikasi Diagnosis Penyakit Jantung ,DOI: 10.32493/jtsi.v7i3.42254

[24] Elmi Ramlan Bugis, Amalia C. Nur’Aidha, Dhananjaya Yama Hudha Kumarajati Alat Monitoring Detak Jantung Portable Menggunakan Sensor Max30102 http://dx.doi.org/10.23960/jitet.v12i3.4504[^26]

[25] Implementasi Machine Learning Untuk Prediksi Penyakit Jantung Menggunakan Algoritma Support Vector Machine Rahmat Hidayat, Yandiko Saputra Sy, Teguh Sujana, Mirdatul Husnah, Haris Tri Saputra, Finanta Okmayura ISSN: 2722-0850

[26] Fandi Aziz Pratama, Afu Ichsan Pradana, Dwi Hartanti Pengembangan Aplikasi Mobile Untuk Monitoring Detak Jantung, Saturasi Oksigen Darah, Dan Suhu Tubuh Dengan Integrasi Iot Menggunakan Esp32 P-ISSN : 2460-2108 E-ISSN : 2620-5181

[27] Trisya Septiana, Mona Arif Muda, Deny Budiyanto, Mahendra Pratama, Winaldi Putra Jaya Analisis Penggunaan Support Vector Machine pada Deteksi Dini Penyakit Diabetes Melitus https://doi.org/10.54082/jupin.643[^30]

[28] Risa Adinda Putri1,YeshaDaniela Aedo,Indra Wijaya,Muhammad Roihan Jannatun Adhen,Dicky Pratama Analisis Implementasi Internet Of Thing Dalam Bidang Kesehatan: Systematic Literature Review Vol 1 No 1 (2024): JREIN - Jurnal Rekayasa Informatika

[29] Amanda Iksanul Putri, Yulia Syarif, Puguh Jayadi, Fadlan Arrazak, Febi Nur Salisah Implementasi Algoritma Decision Tree dan Support Vector Machine(SVM)untuk Prediksi Risiko Stuntingpada Keluarga ISSN(P): 2797-2313 | ISSN(E): 2775-8575

[30] Shafira Amanda Putri, Nabilah Selayanti, Mirechelin Kristanaya, Melinda Putri Azzahra, Muhammad Ghinan Navsih, Kartika Maulida Hindrayani Penerapan Machine Learning Algoritma Random Forest Untuk Prediksi Penyakit Jantung https://doi.org/10.33005/senada.v4i1.376[^36]