

## **Analisis Komparasi Algoritma SVM, Random Forest dan MLP-NN Untuk Klasifikasi Intrusi Perimeter Berbasis Getaran**

Regi Saputra<sup>(1)</sup>, Ari Purno Wahyu Wibowo<sup>(2)</sup>

Universitas Widyatama,  
Jl. Cikutra No.204A, Sukapada, Kec. Cibeunying Kidul, Kota Bandung, Indonesia

Email: <sup>1</sup>saputra.regi@widyatama.ac.id, <sup>2</sup>ari.purno@widyatama.ac.id

---

### **Tersedia Online di**

<http://www.jurnal.unublitar.ac.id/index.php/briliant>

---

### **Sejarah Artikel**

Diterima 22 Mei 2025  
Direvisi 05 Juni 2025  
Disetujui 05 Februari 2026  
Dipublikasikan 22 Februari 2026

---

### **Keywords:**

*Intrusion, Machine Learning, MLP-NN, Random Forest, SVM*

---

---

### **Kata Kunci:**

*Intrusi, Machine Learning, MLP-NN, Random Forest, SVM*

---

### **Corresponding Author:**

Name:  
Regi Saputra  
Email:  
[saputra.regi@widyatama.ac.id](mailto:saputra.regi@widyatama.ac.id)

---

**Abstract:** *The need for perimeter security systems is increasingly important in facing the increasing risk of intrusion to various infrastructures. This study aims to compare the performance of the Support Vector Machine (SVM), Random Forest, and Multi-Layers Perceptron Neural Network (MPL-NN) classification algorithms in separating intrusion and non-intrusion data classes recorded in the SW-420 vibration sensor installed on the perimeter fence. The Message Queuing Telemetry Transport (MQTT) communication protocol is used to connect the sensor to the program that records the dataset. Data collection is carried out through simulations of various vibration scenarios, such as intrusion attempts (intrusion) and environmental disturbances (non-intrusion). Normalization and label encoding techniques are applied to help the algorithm read important features in each data point. The results of the study show that of the three algorithms, Random Forest has a higher accuracy value with a value of 97% followed by the MLP-NN Tanh activation algorithm with an accuracy value of 93%. While the SVM algorithm with the RBF kernel has a value of 90.5%. This means that the Random Forest algorithm has good performance in categorizing vibrations.*

**Abstrak:** Kebutuhan akan sistem keamanan perimeter semakin penting dalam menghadapi risiko penyusupan yang terus meningkat terhadap berbagai infrastruktur. Penelitian ini bertujuan mengkomparasi performa algoritma klasifikasi *Support Vector Machine (SVM)*, *Random Forest*, dan *Multi-Layers Perceptron Neural Network (MPL-NN)* dalam memisahkan kelas data intrusi dan non-intrusi yang terekam dalam sensor getar SW-420 yang dipasang di pagar perimeter. Protokol komunikasi *Message Queuing Telemetry Transport (MQTT)* digunakan guna menghubungkan sensor dengan program yang merekam *dataset*. Pengumpulan data dilakukan melalui

simulasi berbagai skenario getaran, seperti upaya penyusupan (intrusi) dan gangguan lingkungan (non-intrusi). Teknik normalisasi dan *label encoding* diterapkan untuk membantu algoritma dalam membaca fitur penting di setiap data poin. Hasil penelitian menunjukkan dari tiga algoritma tersebut *Random Forest* memiliki nilai akurasi yang lebih tinggi dengan nilai 97% disusul algoritma *MLP-NN* aktivasi *Tanh* dengan nilai akurasi 93%. Sedangkan algoritma *SVM* dengan *kernel RBF* memiliki nilai 90,5%. Artinya algoritma *Random Forest* memiliki performa yang baik dalam mengkategorikan getaran.

## **PENDAHULUAN**

Perlindungan sebuah infrastruktur menjadi kebutuhan yang penting dalam menghadapi ancaman keamanan yang dapat terjadi. Pengawasan terhadap suatu area sangat diperlukan baik untuk sebuah gedung tinggi, kawasan luas, bahkan untuk area rumah sekalipun (Pitafi et al.,

2023). Berdasarkan data dari Pusat Informasi Kriminal Nasional di Kepolisian Republik Indonesia, sepanjang tahun 2024 terjadi 44,726 kasus pencurian dimana mayoritas kasus ini terjadi di rumah, perkantoran, dan tempat komersil. Kendala dalam mengawasi area tersebut adalah kemampuan pengawasan yang tidak dapat bekerja secara 24 jam terus menerus. Bahkan untuk kawasan yang lebih besar tidak dapat hanya mengandalkan tim petugas keamanan saja dalam mengawasinya. Diperlukan sebuah sistem yang mampu mendeteksi potensi terjadinya intrusi atau penyusupan. Titik yang dapat diaplikasikan adalah titik perimeter yang menjadi akses utama untuk masuk ke suatu area (Rizki et al., 2024).

Salah satu opsi yang dapat dimanfaatkan untuk mendeteksi potensi intrusi adalah dengan memanfaatkan sensor getaran seperti sensor *SW-420*. Sensor ini mampu mendeteksi getaran yang terjadi di pagar perimeter (Alamsyah et al., 2023). Namun data getaran dari tempat terbuka dapat dipengaruhi oleh berbagai faktor seperti angin, hujan, maupun kendaraan yang melintas. Sehingga dibutuhkan metode yang tepat untuk membedakan getaran yang berasal dari aktivitas intrusi dan non-intrusi. Dengan berkembangnya teknologi saat ini, maka sistem pengawasan ini pun dapat memanfaatkan fitur mesin pembelajaran (*machine learning*) dalam penerapannya. Pada sistem ini model mesin pembelajaran dapat digunakan untuk menganalisa pola getaran yang terjadi pada sensor getar yang digunakan (Mi et al., 2021).

Permasalahan utama yang diangkat dalam penelitian ini adalah bagaimana membangun model klasifikasi yang mampu membedakan getaran intrusi dan non-intrusi dengan tingkat akurasi yang tinggi. Data getaran intrusi dan non-intrusi dikumpulkan melalui sensor *SW-420* yang terhubung dengan *microcontroller Node MCU ESP8266* dan *dataset* direkam oleh program melalui komunikasi protokol *MQTT*. Lalu *dataset* yang sudah terkumpul dianalisa menggunakan algoritma klasifikasi *Support Vector Machine (SVM)*, *Random Forest*, dan *Multi-layers Perceptron Neural Network (MLP-NN)*.

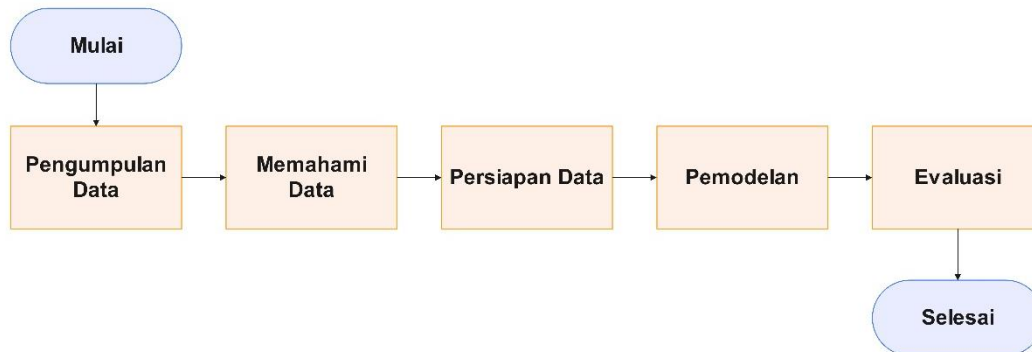
Tujuan dari penelitian ini adalah bagaimana komparasi dari performa tiga algoritma klasifikasi sebelumnya dalam membaca pola getaran yang dihasilkan dari sensor getar *SW-420*. Ruang lingkup penelitian dibatasi dengan pemasangan sensor *SW-420* di pagar tembok dan dilakukan skenario yang menggambarkan keadaan intrusi seperti memanjat pagar dan upaya merusak sensor serta kondisi getaran hujan, angin, dan kendaraan melintas yang terekam. Penelitian ini pun hanya berfokus pada sensor getar dan tidak membahas integrasi dengan alat pendeteksi lain seperti kamera atau alarm lain.

Penelitian ini mengacu pada penelitian sebelumnya yang membahas tentang optimasi pada sistem keamanan perimeter. Penelitian pertama yang penulis temukan adalah penelitian yang dilakukan oleh Shahneela Pitafi (2023) yang membahas bagaimana membuat sebuah sistem intrusi perimeter yang menggunakan beberapa parameter deteksi seperti video, getaran dan suhu. Lalu data diklasterkan menggunakan algoritma *ST-DBSCAN* dan diklasifikasikan dengan algoritma *K-NN classification* (Pitafi et al., 2023). Penelitian selanjutnya adalah penelitian yang dilakukan oleh Qiushi Mi (2021) yang membahas bagaimana kinerja dari algoritma *preprocessing* dan *Multi-layers Perceptron Neural Network* dalam merekognisi pola getaran yang terjadi pada sensor getar *fiber optic* di perimeter. Pada penelitiannya dihasilkan bahwa pola getaran yang terjadi pada sensor dapat direkognisi dan dapat diklasifikasikan oleh algoritma tersebut dengan nilai akurasi 97,6% (Mi et al., 2021). Penelitian selanjutnya adalah penelitian dari Kun Li (2025) yang membahas metode klasifikasi sinyal getaran pada sistem keamanan perimeter berbasis kabel optik  $\phi$ -*OTDR (phase-sensitive Optical Time Domain Reflectometer)*. Peneliti mengembangkan metode kombinasi antara *EMD (Empirical Mode Decomposition)* algoritma *CNN (Convolutional Neural Network)* dan *LSTM (Long Short-Term Memory)*. Hasil eksperimen menunjukkan akurasi klasifikasi sinyal intrusi sebesar 97.3% terhadap tiga jenis pola getaran berbeda (Li et al., 2025).

Hasil dari penelitian ini diharapkan dapat mengembangkan sistem keamanan perimeter berbasis getaran yang lebih akurat. Manfaat yang ingin dicapai adalah memberikan solusi deteksi intrusi yang lebih terjangkau dengan sensor sederhana dan memiliki akurasi tinggi. Selain itu penelitian perbandingan algoritma ini diharapkan dapat menambah studi pada keamanan perimeter berbasis sensor untuk kedepannya.

## METODE

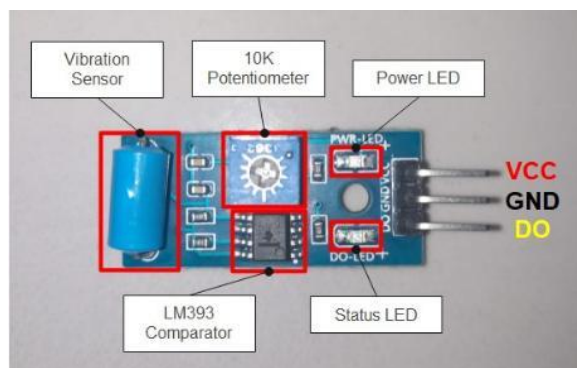
Metode penelitian digambarkan pada *flowchart* Gambar 1. Pada bagian ini dijelaskan bagaimana metode komparasi setiap algoritma *machine learning* dilakukan. Ada beberapa tahapan proses pada metode penelitian yang dilakukan, dimulai dari proses Pengumpulan Data, Memahami Data, Persiapan Data, Pemodelan, hingga Evaluasi (Chapman et al., 2000).



Gambar 1. Flowchart Metode Penelitian yang Dilakukan

### Pengumpulan Data (*Data Collecting*)

Data yang digunakan pada proses penelitian ini adalah data yang diambil dari sensor *SW-420*. Sensor dipasang pada permukaan pagar pembatas seperti yang ditampilkan pada Gambar 4. Komponen sensor *SW-420* yang ditunjukkan pada Gambar 2, terdiri dari komponen sensor yang menangkap getaran pada alat. Lalu ada komponen *LM393* yang bertugas untuk membandingkan voltase sensor dengan input getaran yang masuk pada sensor. Komponen *LM393* ini terhubung juga dengan *potentiometer* yang bertujuan untuk melakukan penyesuaian terhadap nilai getaran yang terbaca. Sensor menghasilkan sinyal digital yang kemudian dikonversi menjadi nilai analog melalui modul *ADC (Analog-to-Digital Converter)*(Alamsyah et al., 2023).



Gambar 2. Komponen Sensor *SW-420*

Proses pengambilan data dilakukan oleh sensor *SW-420* yang terhubung ke *microcontroller NodeMCU ESP8266* dan *battery 9V* sebagai sumber daya listriknya. Pada topologi komunikasinya, sistem ini akan memanfaatkan protokol *Message Queuing Telemetry Transport (MQTT)* dalam mengirimkan data sensor kepada server yang menyimpan dataset. *MQTT* dikenal sebagai protokol yang dapat bekerja sangat baik dalam *bandwidth* yang terbatas(Khan et al., 2021). *Microcontroller* akan bertindak sebagai *MQTT Publisher* yang mengirimkan data melalui koneksi *wifi* kepada *broker MQTT*. Komputer akan bertindak sebagai *broker MQTT* sekaligus menjalankan program *MQTT Subscriber* yang menerima data dari sensor lalu menyimpan setiap data yang tersimpan menjadi *dataset* dalam *file* dengan format *CSV*.

Pengambilan data pada *dataset* ini akan dilakukan dengan melakukan input pada awal program. Pilihan input intrusi atau non-intrusi akan dilakukan saat data akan direkam. Aktivitas

data intrusi meliputi aktivitas memanjat pagar dan percobaan pengrusakan terhadap sensor yang terpasang di pagar pembatas. Sedangkan aktivitas non-intrusi meliputi getaran yang terjadi karena hujan deras maupun ringan, kendaraan yang melintas di dekat sensor, dan getaran-getaran lain yang terjadi diluar aktivitas getaran intrusi.

### Memahami Data (*Data Understanding*)

*Dataset* berisikan akumulasi getaran yang terjadi selama 5 detik dengan 10 nilai getaran yang terekam di setiap 500 ms pada program *subscriber MQTT*. Pada tahap ini dilakukan eksplorasi awal terhadap dataset yang telah dikumpulkan. Visualisasi akan dilakukan menggunakan grafik garis (*line chart*) untuk melihat pola getaran antara getaran intrusi dan non-intrusi. Selain itu, akan digunakan visualisasi dari *correlation matrix* untuk mengetahui hubungan antar fitur getaran 1 hingga 10. *Correlation Matrix* banyak digunakan dalam jurnal-jurnal ilmiah sebagai parameter untuk menilai hubungan antar data yang ada (Nur Fauzi et al., 2025). Tujuan tahap ini adalah untuk mengetahui distribusi dan pola awal yang dapat dijadikan dasar pada proses klasifikasi oleh algoritma.

### Persiapan Data (*Data Preparation*)

Pada tahap *data preparation* akan dilakukan beberapa proses persiapan data agar memastikan kualitas data pada saat dilakukan pemodelan. Nilai yang dihasilkan oleh sensor getar SW-420 memiliki nilai skala yang cukup besar. Agar nilai yang dijadikan dalam pemodelan lebih seimbang maka diperlukan proses normalisasi pada proses persiapannya. Metode normalisasi yang digunakan pada penelitian ini adalah metode *Standard Scaler* (de Amorim et al., 2022).

*Standard Scaler* adalah teknik normalisasi yang menggunakan *z-score normalization*. Nilai pada setiap data dikurangi dengan nilai rata-rata pada kolom yang sama, lalu hasilnya dibagi dengan nilai standar deviasi. Standar deviasi dirumuskan pada Persamaan (1) dan normalisasinya dirumuskan pada Persamaan (2). Proses normalisasi ini telah mengubah nilai *value* getaran menjadi lebih seragam dan menghilangkan perbedaan nilai yang terlalu tinggi yang berpotensi menjadi *outlier*.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad (1)$$

$$z = \frac{(x_i - \bar{x})}{\sigma} \quad (2)$$

Data pada kolom label masih memiliki nilai “intrusi” dan “non-intrusi”. Data tersebut perlu dirubah ke dalam tipe numerik menggunakan teknik *Label Encoder* untuk memudahkan model dalam menentukan klasifikasinya. *Label Encoder* adalah teknik mengubah nilai pada sebuah data menjadi nilai numerik yang dapat memudahkan algoritma machine learning dalam memproses data. Pada proses *encode*, data intrusi dirubah menjadi 0 dan data non-intrusi menjadi 1 (Herdian et al., 2024).

Data yang memiliki nilai duplikasi dihapus dalam dataset agar model dapat bekerja optimal. Distribusi data antara intrusi dan non-intrusi ditunjukkan pada Gambar 8. Data ini memperlihatkan bahwa data intrusi dan non-intrusi terbagi kedalam 2 kelas data yang punya jumlah yang cukup seimbang. Jumlah yang seimbang ini dapat memudahkan model untuk lebih mengenali karakteristik dari setiap datanya.

### Modelling

Proses pembangunan model klasifikasi pada penelitian akan menggunakan tiga algoritma yaitu *Support Vector Machine (SVM)*, *Random Forest*, dan *Multi-layer Perceptron Neural Network (MLP-NN)*. *SVM* adalah algoritma *machine learning* yang biasa digunakan dalam klasifikasi data dengan membuat batas pemisah atau *hyperlane* terhadap masing-masing kelas klasifikasi (Putro, 2024). Pada penelitian ini akan digunakan *kernel* yang umum pada model *SVM* yaitu *kernel linear*, *polynomial*, *RBF* dan *sigmoid* (Zhang et al., 2022). Persamaan *kernel linear*,

*polynomial*, *RBF* dan *sigmoid* ditunjukkan pada persamaan (3), (4), (5) dan (6) (Aryo Anggoro & Permatasari, 2023).

$$K(x_i, x_j) = x_i \cdot x_j \quad (3)$$

$$K(x_i, x_j) = (x_i \cdot x_j + c)^d \quad (4)$$

$$K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2) \quad (5)$$

$$K(x_i, x_j) = \tanh(\gamma(x_i \cdot x_j) + C) \quad (6)$$

*Random Forest* adalah algoritma jenis *ensemble* yang terdiri dari beberapa *decision tree* (Breiman, 2001). Setiap *decision tree* akan mengambil *subset* data dengan metode *bagging* (*bootstrap aggregating*). Dan pada hasil akhirnya, algoritma ini akan melakukan voting terhadap setiap hasil *decision tree* untuk menentukan apakah data termasuk pada klasifikasi intrusi atau non-intrusi. Algoritma ini memiliki keunggulan yang tidak terlalu sensitif terhadap *outlier*. Algoritma ini pun punya performa yang sangat tinggi jika digunakan dalam data yang besar (Susetyoko et al., 2023). Algoritma *Random Forest* banyak digunakan dalam beberapa penelitian karena kemampuannya dalam mengurangi *overfitting* terhadap data (Hassan et al., 2022). Persamaan algoritma *Random Forest* secara sederhana dapat ditunjukkan pada Persamaan (7).

$$y = \text{mode}(h_1(x), h_2(x), \dots, h_n(x)) \quad (7)$$

*Multi-layer Perceptron (MLP-NN)* adalah algoritma klasifikasi yang ada pada *machine learning* dengan menggunakan konsep saraf tiruan atau *neuron* (Popescu & Balas, 2009). *MLP-NN* terdiri dari beberapa lapisan *neuron* tiruan yang disebut sebagai *perceptron*. *Input* pada *perceptron layer* pertama akan menghasilkan *output* yang dijadikan *input* untuk *perceptron* pada *layer* berikutnya atau biasa disebut *hidden layer*, hingga pada hasil *output* terakhir. Pada setiap *perceptron* yang memproses data ada fungsi aktivasi yang berjalan yang bertugas untuk menentukan nilai *output* dikeluarkan. Pada metode klasifikasi *MLP-NN* ada tiga fungsi aktivasi yang umum digunakan yaitu aktivasi *ReLU (Rectified Linear Unit)*, *sigmoid/logistic*, dan *tanh* (Abdulrahman Safar et al., 2023). Persamaan setiap *perceptron* dalam algoritma *MLP-NN* digambarkan dalam Persamaan (8).

$$\alpha^{(l)} = f^{(l)}(W^{(l)}\alpha^{(l-1)} + b^{(l)}) \quad (8)$$

Data akan dibagi menggunakan *train\_test\_split* dengan rasio 80% *data training* dan 20% *data test* serta parameter *stratify* untuk menjaga proporsi kelas. Teknik *5-fold cross validation* digunakan dalam proses pelatihan model untuk menghindari *overfitting*. Dengan membagi data menjadi lima *subset* dan melakukan validasi silang, performa model diuji pada berbagai variasi data. Hasil rata-rata dari lima pengujian ini menggambarkan kemampuan model dalam melatih data training nya (Rodrigues et al., 2025).

## Evaluasi

Hasil evaluasi dari pemodelan dapat diukur dari beberapa parameter. Umumnya parameter pengukuran dilakukan dengan mengkalkulasi nilai *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, dan *False Negative (FN)* (Chicco & Jurman, 2020). Hubungan antar parameter tersebut digambarkan dalam hubungan *confusion matrix* yang menunjukkan kondisi label yang benar (*True Label*) dengan label prediksi (*Prediction Label*) seperti pada Tabel 1.

Tabel 1. Confusion Matrix

		<i>Prediction Label</i>	
		<i>Positive</i>	<i>Negative</i>
<i>True Label</i>	<i>Positive</i>	TP	FN
	<i>Negative</i>	FP	TN

Dari parameter tersebut, diambil empat kriteria penilaian dalam menghitung performa model setiap algoritma *machine learning* yang terdapat pada fungsi *Classification Report* yaitu

*Precision*, *Recall*, *F1-Score* dan *Accuracy* (Lohani et al., 2022). Presisi adalah rasio prediksi positif yang terbaca benar terhadap total nilai prediksi positif. *Recall* adalah banyaknya pembacaan positif dari semua data positif yang ada. *F1-Score* adalah penggabungan antara nilai Presisi dan *Recall* dan melihat keseimbangan dari dua nilai tersebut. Sedangkan *Accuracy* adalah rasio prediksi yang benar dengan total semua data prediksi yang dilakukan (Khan et al., 2021). *Precisi*, *Recall*, *F1-Score* dan *Accuracy* dirumuskan dalam persamaan (9), (10), (11), dan (12)

$$Presisi = \frac{TP}{TP+FP} \quad (9)$$

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (11)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

Metode *ROC-AUC* (*Receiver Operating Characteristic – Area Under Curve*) digunakan juga pada penelitian ini untuk mengukur hasil evaluasi model dalam membedakan kedua kelas intrusi dan non-intrusi berdasarkan nilai probabilitas. Kurva *ROC* menunjukkan proporsi data non-intrusi yang salah diklasifikasikan. Sedangkan untuk *AUC* mengukur luas area dibawah kurva *ROC* di rentang nilai 0 sampai 1. Nilai mendekati 1 menunjukkan model memiliki kemampuan yang baik dalam klasifikasi (Prakoso & Hermawan, 2023).

## HASIL DAN PEMBAHASAN

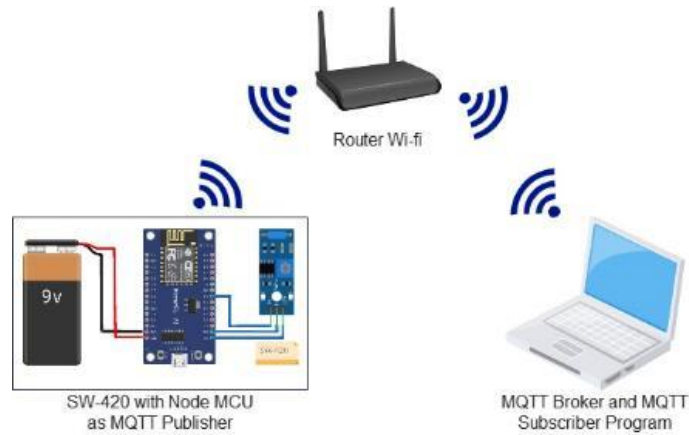
### Hasil Pengumpulan Data

Alat sensor dan *microcontroller* yang sudah dirancang ditunjukkan pada Gambar 3. Sensor yang sudah dirakit dan terhubung dengan *microcontroller* dipasang di pagar perimeter untuk menangkap getaran. Selama pengujian sensor dapat mendeteksi perubahan getaran yang terjadi secara responsif dan *real-time*. Hal ini menunjukkan sensor *SW-420* bekerja cukup baik dalam membaca getaran yang diterima (Alamsyah et al., 2023; Ghazali & Rahiman, 2022).



Gambar 3. Sensor dan *Microcontroller*.

Topologi yang diterapkan dalam proses pengumpulan data ditunjukkan pada Gambar 4. Data pembacaan sensor berhasil ditangkap oleh aplikasi *MQTT subscriber* melalui *MQTT broker*. Selama pengujian proses pengiriman data berjalan stabil jika koneksi *wifi* memiliki sinyal yang baik. Topologi ini menunjukkan bahwa protokol *MQTT* mampu mengirimkan data getaran secara *real-time* kepada server (Hairatunnisa et al., 2021; Khan et al., 2021).



Gambar 4. Topologi Sensor, Microcontroller, dan Program Pengambilan Dataset Dengan Protokol MQTT

Sensor diujicoba selama beberapa hari guna menangkap skenario data intrusi dan non-intrusi. Aktivitas skenario intrusi dan non-intrusi ditunjukkan pada Gambar 5 dan Gambar 6. Setiap skenario data intrusi dan non-intrusi yang masuk diperhatikan agar kondisi yang terekam optimal dan tidak ada kondisi sensor *idle* yang memungkinkan data memiliki *noise*.



Gambar 5. Skenario Aktivitas Intrusi



Gambar 6. Aktivitas Non-Intrusi (Hujan dan Kendaraan Melintas)

Tabel 2 menunjukkan data yang berhasil direkam oleh kedalam file *dataset*. Sistem berhasil merekam 1025 data getaran. Data getaran yang terekam terdiri dari 519 data getaran intrusi dan 506 data getaran non-intrusi.

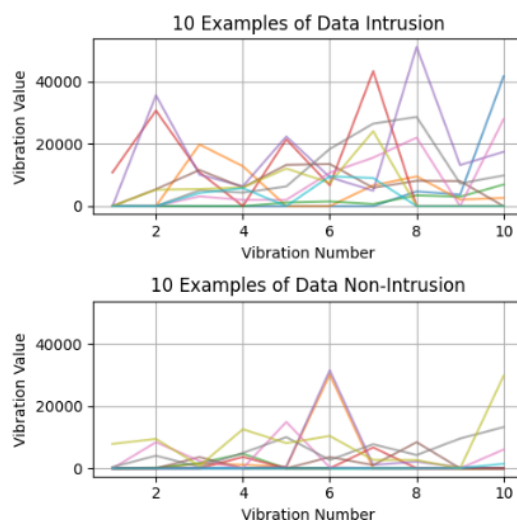
Tabel 2. Tabel Data Intrusi dan Non-Intrusi

ID	Nilai Getaran										Label
	1	2	3	4	5	6	7	8	9	10	
0	0	0	0	0	0	0	0	0	0	0	Non-intrusi
1	0	0	1584	1199	465	3018 7	0	0	0	0	Non-intrusi

2	0	170	1704	4748	0	0	0	0	0	0	Non-intrusi
3	0	0	0	0	0	0	0	0	0	0	Non-intrusi
4	0	0	0	3686	26	0	6768	0	0	0	Non-intrusi
.....											
1020	4671	1016 4	1177 5	9109	9438	8979	0	0	0	0	Intrusi
1021	0	9723	3118 2	0	0	0	0	0	0	0	Intrusi
1022	9920	9801	9455	0	1529 2	1628 8	8938	9131	2066 4	0	Intrusi
1023	0	0	0	2687 6	9485	1430 9	1850 9	0	0	0	Intrusi
1024	0	0	8820	0	1776 2	2291 5	1988 1	0	0	0	Intrusi

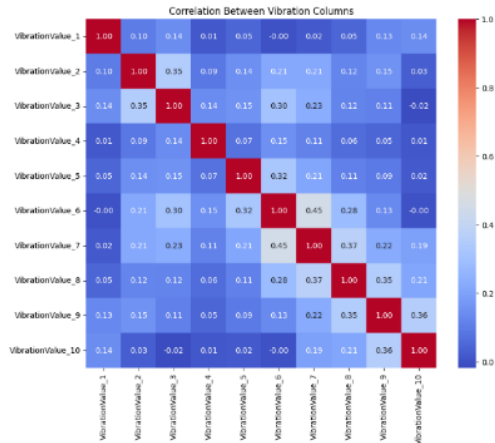
### Visualisasi Data

Visualisasi pada Gambar 7 menampilkan 10 contoh pada masing-masing data intrusi dan non-intrusi. Pada visualisasi tersebut getaran yang terjadi karena intrusi nilainya cenderung tinggi dan fluktuatif. Sedangkan getaran yang terjadi pada kategori non-intrusi lebih kecil fluktuasinya.



Gambar 7. Visualisasi Contoh Getaran Intrusi dan Non-Intrusi

Setiap nilai getaran yang terekam dalam *dataset* memiliki korelasi antara satu dengan yang lainnya yang dapat saling memengaruhi pada saat proses klasifikasi. Korelasi setiap datanya divisualisasikan oleh *Correlation Matrix* seperti pada Gambar 8. Nilai yang mendekati angka 1 ada pada kolom *VibrationValue\_6* dan *VibrationValue\_7* menandakan bahwa korelasi antara kedua nilai getaran yang memiliki hubungan yang kuat (Jasman et al., 2022; Nur Fauzi et al., 2025).



Gambar 8. Correlation Matrix Setiap Kolom

### Hasil Persiapan Data

Proses normalisasi menggunakan *Standard Scaler* berhasil dilakukan terhadap seluruh nilai getaran dari *VibrationValue\_1* hingga *VibrationValue\_10*. Hasil dari proses ini menunjukkan bahwa distribusi nilai pada setiap kolom fitur telah berada pada skala yang lebih seragam, dengan rata-rata mendekati nol dan standar deviasi satu seperti yang ditunjukkan oleh Gambar 9 (de Amorim et al., 2022; Elkhadir & Achkari Begdouri, 2025). Dengan skala data yang tidak terlalu jauh, model klasifikasi dapat mengenali lebih baik pola getaran antara intrusi dan non-intrusi.

```
array([[ -0.29429445, -0.45211235, -0.57020734, ..., -0.47230345,
        -0.48411343, -0.43779345],
       [ -0.29429445, -0.45211235, -0.4493073 , ..., -0.47230345,
        -0.48411343, -0.43779345],
       [ -0.29429445, -0.43254432, -0.4401482 , ..., -0.47230345,
        -0.48411343, -0.43779345],
       ...,
       [ 0.96357261,  0.67604212,  0.15145294, ...,  0.33949723,
        2.3560526 , -0.43779345],
       [ -0.29429445, -0.45211235, -0.57020734, ..., -0.47230345,
        -0.48411343, -0.43779345],
       [ -0.29429445, -0.45211235,  0.10298606, ..., -0.47230345,
        -0.48411343, -0.43779345]])
```

Gambar 9. Data Setelah Normalisasi

Setelah proses normalisasi dilakukan teknik *Label Encoder* pada kolom “*Label*”. Nilai kolom yang awalnya berupa nilai string “intrusi” dan “non-intrusi” telah berhasil dikonversi ke bentuk numerik 0 dan 1 seperti ditunjukkan pada Gambar 10. Hasil *encoding* ini memungkinkan seluruh algoritma klasifikasi untuk memproses label dalam format numerik (Herdian et al., 2024).



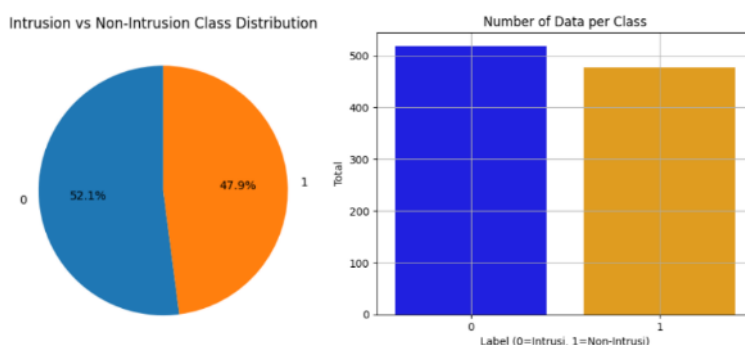
Gambar 10. Hasil Label Encode

Pemeriksaan duplikasi data pun dilakukan pada tahap ini. Terdapat 28 data yang memiliki kesamaan nilai yaitu 0 di setiap *VibrationValue* yang terekam. Nilai itu terjadi ketika sistem tidak mendeteksi getaran pada pagar perimeter. Setelah penghapusan duplikat, jumlah data yang dapat digunakan untuk pemodelan berjumlah 997 dari total awal 1025 seperti ditunjukkan Gambar 11. Penghapusan ini penting dilakukan agar tidak terjadi pengulangan pola data yang sama.

VibrationValue_1	997
VibrationValue_2	997
VibrationValue_3	997
VibrationValue_4	997
VibrationValue_5	997
VibrationValue_6	997
VibrationValue_7	997
VibrationValue_8	997
VibrationValue_9	997
VibrationValue_10	997
Label	997

Gambar 11. Jumlah Data Setelah Menghapus Data Duplikat

Analisis distribusi kelas dilakukan untuk melihat keseimbangan jumlah data antara data intrusi dan non-intrusi. Visualisasi berupa *pie chart* dan *bar chart* digunakan untuk menggambarkan distribusi data seperti pada Gambar 12. Gambar menunjukkan distribusi yang relatif seimbang antara data intrusi dan non-intrusi yaitu sekitar 52,1% data intrusi dan 47,9% data non-intrusi. Keseimbangan ini memberikan keuntungan bagi pemodelan karena resiko bias klasifikasi terhadap salah satu kelas menjadi kecil (Jasman et al., 2022).



Gambar 12. Distribusi Data Intrusi dan Non-Intrusi

### Hasil Modelling

Data pada Tabel 3 menampilkan hasil dari skor rata-rata *cross validation* dengan waktu eksekusi model di setiap algoritma beserta parameterinya. Berdasarkan data pada tabel tersebut, model klasifikasi dari algoritma *Random Forest* memiliki nilai rata-rata *cross validation* yang paling baik serta memiliki waktu eksekusi yang cukup cepat dibandingkan dengan dua algoritma lainnya (Hassan et al., 2022; Susetyoko et al., 2023). Algoritma *SVM* yang memiliki waktu eksekusi tercepat dibandingkan dua algoritma lain namun nilai rata-rata *cross validation* masih dibawah algoritma *Random Forest*. Algoritma *MLP-NN* memiliki waktu eksekusi yang lebih lama diantara semua algoritma karena proses modelling memproses data *training* dengan teknik *cross validation* ditambah dengan parameter *max\_iter* agar algoritma dapat menghasilkan model yang konvergen. Pada proses *modelling* algoritma *MLP-NN*, fungsi aktivasi *ReLU* dan *tanh* memerlukan dua *hidden layer* yang terdiri dari 100 *perceptron* di *layer* pertama dan 50 *perceptron* di *layer* kedua untuk menghasilkan model yang konvergen. Sedangkan fungsi aktivasi *logistic* memerlukan tiga *hidden layer* yang terdiri dari 100 *perceptron* di *layer* pertama, 100 *perceptron* di *layer* kedua dan 50 *perceptron* di *layer* ketiga untuk menghasilkan model yang konvergen (Intan et al., 2024).

Tabel 3. Skor *Cross Validation* dan Waktu Eksekusi

Algoritma	Parameter (Kernel / Aktivasi)	Skor Cross Validation	Waktu Eksekusi (s)
SVM	Linear	0,83	0,754
	Polynomial	0,64	0,879

	<i>RBF</i>	0,88	0,592
	<i>Sigmoid</i>	0,81	0,501
<b>Random Forest</b>	-	0,95	2,495
	<i>ReLU</i>	0,87	35,026
<b>MLP-NN</b>	<i>Logistic</i>	0,88	21,612
	<i>Tanh</i>	0,87	80,605

## Hasil Evaluasi

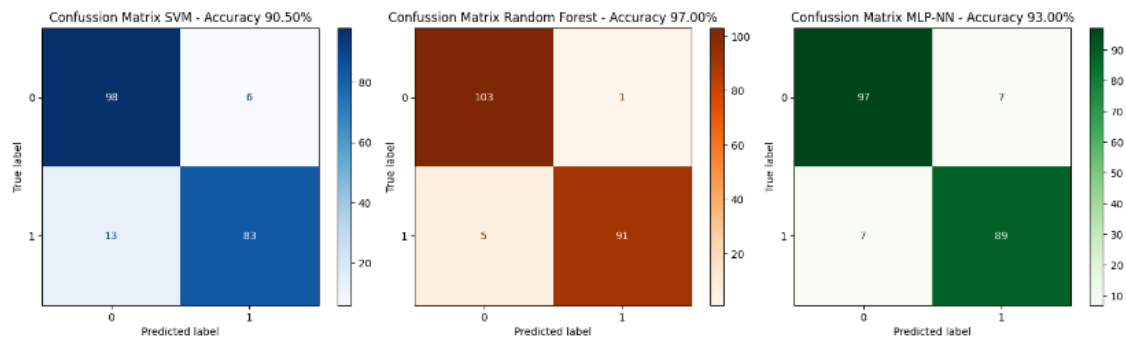
Tabel 4 menampilkan hasil evaluasi masing-masing model berdasarkan penilaian *precision*, *recall*, *f1-score* dan *accuracy*. Algoritma *SVM* diuji dengan empat jenis kernel yang berbeda (*linear*, *polynomial*, *rbf*, dan *sigmoid*) dan algoritma *MLP-NN* diuji dengan tiga jenis fungsi aktivasi (*ReLU*, *logistic*, dan *tanh*).

Tabel 4. Hasil Evaluasi *Modelling*

Algoritma	Parameter (Kernel / Aktivasi)	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy (%)</i>
<b>SVM</b>	<i>Linear</i>	0,86	0,85	0,85	85,5
	<i>Polynomial</i>	0,80	0,69	0,66	68,5
	<i>RBF</i>	0,91	0,91	0,90	90,5
	<i>Sigmoid</i>	0,86	0,86	0,86	86
<b>Random Forest</b>	-	0,97	0,97	0,97	97
<b>MLP-NN</b>	<i>ReLU</i>	0,91	0,91	0,91	91
	<i>Logistic</i>	0,93	0,92	0,92	92
	<i>Tanh</i>	0,93	0,93	0,93	93

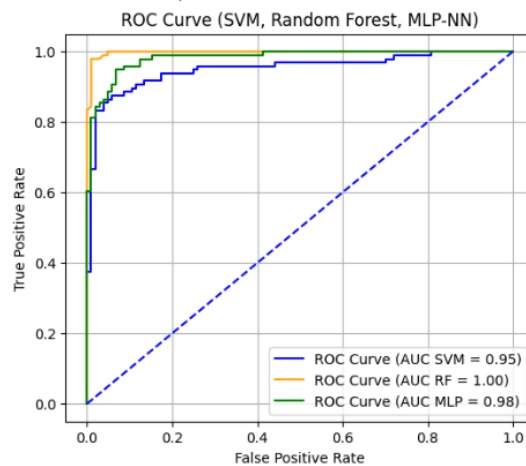
Berdasarkan data pada Tabel 4, algoritma *Random Forest* menunjukkan performa klasifikasi terbaik dengan nilai akurasi 97% serta memiliki waktu eksekusi yang relatif cepat. Hal ini disebabkan oleh keunggulan algoritma *Random Forest* dalam menangani data yang memiliki sifat *non-linear* seperti data getaran pada sensor *SW-420* (Susetyoko et al., 2023). Hasil akurasi tertinggi yang diperoleh dari algoritma *Random Forest* dalam penelitian ini sejalan dengan temuan penelitian lain yang memanfaatkan gabungan beberapa sensor termasuk sensor getaran dan kamera video untuk mendeteksi intrusi (Pitafi et al., 2023). Namun penelitian ini menunjukkan bahwa bahkan tanpa kombinasi alat deteksi lain, sensor *SW-420* dan *Random Forest* tetap menghasilkan akurasi tinggi yang menunjukkan potensinya dalam deteksi getaran intrusi dengan baik. Algoritma *SVM* dengan kernel *RBF* dan algoritma *MLP-NN* dengan fungsi aktivasi *tanh* juga memberikan performa yang cukup baik dengan masing-masing nilai akurasi di 90,5% dan 93%. *Kernel RBF* pada *SVM* memiliki kemampuan untuk memetakan data kedalam dimensi yang lebih tinggi sehingga dari performa pembacaan getaran dapat lebih baik dibandingkan dengan kernel yang lainnya. Sedangkan algoritma *MLP-NN* dengan aktivasi *tanh* memiliki kinerja yang baik sehingga nilai akurasinya lebih tinggi dibandingkan fungsi aktivasi lainnya. Namun dari segi waktu eksekusi, algoritma ini mempunyai catatan waktu yang lebih lama dibandingkan dengan algoritma lainnya. Meski sensor yang digunakan berbeda, penelitian ini berhasil mendekati akurasi dari penelitian lain dengan menggunakan sensor *SW-420* yang lebih ekonomis dan berpotensi menunjukkan efisiensi biaya tanpa mengorbankan performa klasifikasi (Mi et al., 2021).

Visualisasi *confusion matrix* masing-masing algoritma dengan hasil terbaiknya ditampilkan pada Gambar 9 untuk algoritma *SVM kernel RBF*, *Random Forest*, dan *MLP-NN* aktivasi *tanh*. Ketiga model memiliki nilai yang cukup baik namun *Random Forest* menunjukkan kesalahan klasifikasi yang paling rendah dibanding yang lainnya.



Gambar 13. *Confusion Matrix* Hasil Evaluasi

Evaluasi performa modelling juga dilakukan menggunakan *ROC-AUC* dengan tujuan untuk melihat kemampuan model dalam memisahkan data ke dalam kelas intrusi dan non-intrusi. Pada visualisasi Gambar 10 menunjukkan kurva *Random Forest* memiliki kurva lebih tinggi dan nilai *AUC* yang lebih tinggi dibanding performa algoritma yang lainnya. Hasil ini sejalan dengan penelitian lain yang menggunakan evaluasi *ROC-AUC* sebagai acuan dalam melihat performa klasifikasi (Prakoso & Hermawan, 2023).



Gambar 14. Kurva *ROC-AUC*

Penelitian ini memberikan implikasi bahwa kombinasi antara sensor sederhana dan algoritma machine learning tetap mampu menghasilkan sistem klasifikasi yang akurat dalam melakukan deteksi intrusi berbasis getaran. Hasil penelitian ini juga membuka potensi implementasi sistem keamanan perimeter yang lebih hemat biaya namun tetap bagus dalam akurasi yang berlebih untuk di area yang memiliki keterbatasan infrastruktur dan sumber daya manusia. Adapun untuk penelitian selanjutnya, hasil penelitian ini dapat dijadikan dasar dalam pengembangan sistem deteksi intrusi dengan dimensi data dan parameter yang lebih kompleks dan bervariasi.

## SIMPULAN

Kesimpulan dari penelitian ini didapat bahwa ketiga algoritma *SVM*, *Random Forest*, dan *MLP-NN* memiliki nilai akurasi yang sangat baik dalam mengklasifikasikan data getaran yang terjadi akibat intrusi atau non-intrusi. Performa terbaik dihasilkan dari algoritma *Random Forest* yang memiliki nilai akurasi tertinggi. Dengan komponen sederhana seperti sensor *SW-420* dan protokol *MQTT* serta algoritma *Random Forest*, penelitian ini dapat membuktikan bahwa algoritma machine learning dapat mengoptimalkan pembacaan potensi getaran intrusi pada data yang diambil dari pagar perimeter. Saran penulis pada pengembangan penelitian selanjutnya algoritma deteksi intrusi ini dapat digabungkan dengan parameter deteksi lain seperti video pada kamera *CCTV* atau integrasi dengan sistem *IoT* dalam implementasi model *machine learning*-nya.

## DAFTAR RUJUKAN

- Abdulrahman Safar, A., Salih, D. M., & Murshid, A. M. (2023). Pattern recognition using the multi-layer perceptron (MLP) for medical disease: A survey. *Int. J. Nonlinear Anal. Appl*, *14*, 2008–6822. <https://doi.org/10.22075/ijnaa.2022.7114>
- Alamsyah, S. N., Wery Melisa, Sari, O., Mutia Raudhatul Zahra, Muhammad Yuliansyah Putra, Zafran Afif, Shalih Muhammad Abdul Azhim, Yuliza, E., & Ekawita, R. (2023). Distance Range Test of SW-420 Sensor-Based Vibration Detection System. *Jurnal Kumparan Fisika*, *6*(3), 177–184. <https://doi.org/10.33369/jkf.6.3.177-184>
- Aryo Anggoro, D., & Permatasari, D. (2023). Performance Comparison of the Kernels of Support Vector Machine Algorithm for Diabetes Mellitus Classification. *IJACSA) International Journal of Advanced Computer Science and Applications*, *14*(2), 215. [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- Breiman, L. (2001). *Random Forests* (Vol. 45). <https://doi.org/https://doi.org/10.1023/A:1010933404324>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0 Step-by-step Data Mining Guide*.
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1). <https://doi.org/10.1186/s12864-019-6413-7>
- de Amorim, L. B. V., Cavalcanti, G. D. C., & Cruz, R. M. O. (2022). *The choice of scaling technique matters for classification performance*. <https://doi.org/10.1016/j.asoc.2022.109924>
- Elkhadir, Z., & Achkari Begdouri, M. (2025). Enhancing internet of things attack detection using principal component analysis and kernel principal component analysis with cosine distance and sigmoid kernel. *International Journal of Electrical and Computer Engineering (IJECE)*, *15*(1), 1099. <https://doi.org/10.11591/ijece.v15i1.pp1099-1108>
- Ghazali, M. H. M., & Rahiman, W. (2022). Vibration-Based Fault Detection in Drone Using Artificial Intelligence. *IEEE Sensors Journal*, *22*(9), 8439–8448. <https://doi.org/10.1109/JSEN.2022.3163401>
- Hairatunnisa, H., Nugroho, H. A., & Margiono, R. (2021). Analisis Kinerja Protokol MQTT dan HTTP Pada Akuisisi Data Magnet Berbasis Internet of Things. *Jurnal Ilmiah Informatika*, *6*(2), 71–80. <https://doi.org/10.35316/jimi.v6i2.1351>
- Hassan, I. H., Abdullahi, M., Aliyu, M. M., Yusuf, S. A., & Abdulrahim, A. (2022). An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection. *Intelligent Systems with Applications*, *16*. <https://doi.org/10.1016/j.iswa.2022.200114>
- Herdian, C., Kamila, A., & Agung Musa Budidarma, I. G. (2024). Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi. *Technologia: Jurnal Ilmiah*, *15*(1), 93. <https://doi.org/10.31602/tji.v15i1.13457>
- Intan, P. A., Ardhia, I. C., & Sri, C. K. A. (2024). Klasifikasi Penyakit Stunting Menggunakan Algoritma Multi-Layer Perceptron. *Journal MIND Journal | ISSN*, *9*(1), 52–63. <https://doi.org/10.26760/mindjournal.v9i1.52-63>
- Jasman, T. Z., Hasmin, E., Sunardi, Susanto, C., & Musu, W. (2022). Perbandingan Logistic Regression, Random Forest, dan Perceptron pada Klasifikasi Pasien Gagal Jantung. *CSRID (Computer Science Research and Its Development Journal)*, *14*(3), 271–286. <https://doi.org/10.22303/csrid.14.3.2022.271-286>
- Khan, M. A., Khan, M. A., Jan, S. U., Ahmad, J., Jamal, S. S., Shah, A. A., Pitropakis, N., & Buchanan, W. J. (2021). A deep learning-based intrusion detection system for mqtt enabled iot. In *Sensors* (Vol. 21, Issue 21). MDPI. <https://doi.org/10.3390/s21217016>

- Li, K., Zhen, Y., Li, P., Hu, X., & Yang, L. (2025). Optical Fiber Vibration Signal Recognition Based on the EMD Algorithm and CNN-LSTM. *Sensors*, 25(7), 2016. <https://doi.org/10.3390/s25072016>
- Lohani, D., Crispim-Junior, C., Barthélemy, Q., Bertrand, S., Robinault, L., & Rodet, L. T. (2022). Perimeter Intrusion Detection by Video Surveillance: A Survey. In *Sensors* (Vol. 22, Issue 9). MDPI. <https://doi.org/10.3390/s22093601>
- Mi, Q., Yu, H., Xiao, Q., & Wu, H. (2021). Intrusion behavior classification method applied in a perimeter security monitoring system. *Optics Express*, 29(6), 8592. <https://doi.org/10.1364/oe.415929>
- Nur Fauzi, N. P., Khomsah, S., & Putra Wicaksono, A. D. (2025). Penerapan Feature Engineering dan Hyperparameter Tuning untuk Meningkatkan Akurasi Model Random Forest pada Klasifikasi Risiko Kredit. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 12(2), 251–262. <https://doi.org/10.25126/jtiik.2025128472>
- Pitafi, S., Anwar, T., Widia, I. D. M., Yimwadsana, B., & Pitafi, S. (2023). Revolutionizing Perimeter Intrusion Detection: A Machine Learning-Driven Approach with Curated Dataset Generation for Enhanced Security. *IEEE Access*, 11, 106954–106966. <https://doi.org/10.1109/ACCESS.2023.3318600>
- Popescu, M.-C., & Balas, V. E. (2009). Multilayer Perceptron and Neural Networks. *WSEAS Transactions on Circuits and Systems*, 8(7), 579–588.
- Prakoso, C., & Hermawan, A. (2023). Perbandingan Model Machine Learning dalam Analisis Sentimen Ulasan Pengunjung Keraton Yogyakarta pada Google Maps. *Kajian Ilmiah Informatika Dan Komputer*, 4(3), 1292–1302. <https://doi.org/10.30865/klik.v4i3.1419>
- Putro, I. H. (2024). Performance Comparison of SVM Kernels for Intrusion Detection System Using UNSW-NB15 Dataset. *Jurnal Teknik Elektro*, 17(2).
- Rizki, Z., Lbs, F., Sylvia, T., & Nasution, D. (2024). The Design Of Perimeter Intrusion Detection System (PIDS) Surveillance Alarm Using USB Webcam And Artificial Intelligence Based On Web And Telegram Bot At Medan Aviation Polytechnic. *Jurnal Scientia*, 13. <https://doi.org/10.58471/scientia.v13i04>
- Rodrigues, Á. L., Gontijo, F., Niquini, F., Emilio, W., & Moreno, G. (2025). Spatial cross-validation for Machine Learning model estimates. *ResearchGate*. <https://doi.org/10.13140/RG.2.2.34161.19047>
- Susetyoko, R., Purwantini, E., Nur Iman, B., & Satriyanto, E. (2023). An Improved Accuracy of Multiclass Random Forest Classifier with Continuous Attribute Transformation Using Random Percentile Generation. 13(3). <https://doi.org/https://doi.org/10.18517/ijaseit.13.3.18379>
- Zhang, F., Zhen, P., Jing, D., Tang, X., Chen, H. B., & Yan, J. (2022). SVM Based Intrusion Detection Method with Nonlinear Scaling and Feature Selection. *IEICE Transactions on Information and Systems*, E105D(5), 1024–1038. <https://doi.org/10.1587/transinf.2021EDP7184>