



Research Article

Edge-Computing-Based Anomaly Detection of Rotating Machines Using Artificial Neural Networks

Alireza Mostafavi, Aysan Alizadeh and Ali Sadighi*

School of Mechanical Engineering, University of Tehran, Tehran, Iran

* corresponding author: (asadighi@ut.ac.ir)

Article Info

Article history:

Received: 24 December 2024

Revised: 1 June 2025

Accepted: 20 July 2025

Keywords:

Edge computing,
online learning,
anomaly detection,
autoencoders,
rotating machines

Abstract

Manufacturing companies face issues due to demand for high-quality, affordable products. Since maintenance costs account for 60–70% of production costs, real-time fault detection is vital to lower maintenance expenses and extend equipment life. This article introduces a high-performance anomaly detection framework using edge computing for real-time industrial asset monitoring. Hardware and firmware were designed to perform critical tasks such as data acquisition, preprocessing, feature extraction, and algorithm training on the microcontroller unit (MCU), despite limited processing and memory. Using a 3-axis accelerometer for vibration signals, the MCU stores training data in Flash memory. An autoencoder with three hidden layers is trained on the edge device to model normal operating conditions, and reconstruction error of new data detects anomalies. This study is, to the best of our knowledge, the first to train an artificial neural network (ANN) on an MCU for comprehensive edge-based condition monitoring, achieved over 99.9% accuracy when validated on a centrifugal pump.

Introduction

Rotating equipment, such as pumps and turbines, is critical to industry, and their failures can cause significant financial and safety issues. Therefore, methods like Condition Monitoring (CM) and Prognostics and Health Management (PHM) are used to detect faults at an early stage. Traditional maintenance relies on regular inspections, which can be costly and subjective. Recently, predictive maintenance has been gaining popularity through data-driven approaches using artificial intelligence and machine learning [1]. These techniques make use of the vast amount of sensor data available in industrial systems; however, processing and transmitting this data demand substantial communication and computing power.

Cloud-based frameworks [3,4] for remote condition monitoring face challenges related to latency, bandwidth, and scalability [5]. Edge computing addresses these issues by processing data near its source [6], reducing dependence on cloud servers. Although earlier studies [5,7,8] implement

trained models on edge devices, none achieve full on-device full on-device learning, as they rely on prior training on PCs or cloud platforms.

This work presents a self-learning edge device capable of performing full data processing and model training locally. A deep autoencoder trained only on normal data acts as a novelty detector [10–12]. The proposed approach performs unsupervised learning directly on the MCU, unlike previous methods [13–18] that depend on pre-trained networks. This provides a low-cost, standalone, adaptive solution for real-time anomaly detection.

Methodology

Our approach combines feature engineering, a novelty detector, and custom hardware, all optimized for on-edge deployment. Considering MCU limitations, we initially perform feature engineering on raw vibration samples, extracting a total of 87 features from the time, frequency, and time-frequency domains (Tables 1 and 2).

To Cite this article:

Mostafavi, A.R., Alizadeh, A. and Sadighi, A. 2025. Edge-Computing-Based anomaly detection of rotating machines using artificial neural networks, Sharif Mechanical Engineering Journal, 41(2), 33-47.

<https://doi.org/10.24200/j40.2025.65863.1730>

For anomaly detection, we use an autoencoder (Fig. 1), a neural network trained solely on data representing the machine's healthy condition. The network learns to precisely reconstruct normal inputs; therefore, a high Mean Squared Error (MSE) on new, unseen data indicates a fault. We validate this method using a hydraulic test rig (Fig. 2) that simulates three common pump faults: cavitation, a vane tip fault (Fig.3), and an impeller crack (Fig. 4). The final device hardware (Fig.12) comprises an STM32H743ZI2 microcontroller with 1 MB RAM, an accelerometer, an XBee wireless module, and two lithium-ion batteries.

Training the network within the MCU's 1 MB RAM limit is the main implementation challenge on the edge device. We address this with a batch-based learning approach. Once installed, the device begins a learning phase, collecting 400 healthy samples. Instead of storing all data at once, it trains the network sequentially using four batches of 100 samples each. The memory allocated to each batch is overwritten immediately after processing, allowing the network to train on the entire dataset without exceeding RAM limits. Afterward, the device switches to monitoring mode. It periodically collects new data, computes the 87 features, and calculates the MSE. If the error remains normal, the device remains in low-power mode. If the MSE exceeds the pre-set anomaly threshold, it sends an alarm via the XBee module for operator review.

Results and Discussion

We first confirmed the effectiveness of our feature engineering; while the raw signals for healthy and faulty states look similar (Fig. 5), the extracted features, especially Wavelet Packet Decomposition (WPD), showed clear distinctions (Fig. 6). We then optimized the autoencoder architecture offline. A three-hidden-layer network performed best (Fig. 7). After optimizing the learning rate (Fig. 8) and dataset size (Fig. 9), we selected the final architecture detailed in Table 3. We validated the batch-based training on a PC. As shown in Fig. 10, the training loss remained stable across four batches. This PC-trained model, tested on 2800 faulty and 600 unseen standard samples, achieved over 99.9%

performance scores (Table 5). Fig. 11 and the confusion matrix in Table 4 displayed a near-perfect separation of faulty and healthy data.

After verifying the architecture, the complete algorithm was transferred to the STM32H743ZI2 microcontroller. The device successfully carried out on-edge data acquisition, feature extraction, training, and anomaly detection (Fig. 13). Although training took longer on the MCU (466s compared to 3s on the PC, Table 6), the resulting model achieved near-perfect classification metrics, with precision, recall, and accuracy all equal to 1. The MSE distribution (Fig. 14) distinguished healthy (<3) from faulty samples ($\text{MSE} > 500$), demonstrating high sensitivity to abnormal behavior.

These results confirmed that the proposed system can independently learn and detect faults directly on the edge device without external computing support. Despite the MCU's limited processing power, it achieved accuracy comparable to that of offline training.

Conclusion

This paper presents a fully edge-based anomaly detection system for rotating machines that performs complete neural network training and inference on a microcontroller. Our framework enables on-device training using a batch-based method. This unsupervised online learning approach effectively overcomes key limitations of existing systems. It removes the need for unavailable faulty data and avoids the costly latency associated with cloud computing. Additionally, because training occurs on the edge, the device is highly adaptable and not limited to a single machine, unlike offline-trained models. We developed a lightweight autoencoder and deployed the entire model, including its custom training algorithm, onto a low-cost, low-power MCU. The device can learn the unique healthy behavior of any attached rotating machine. Validation on a centrifugal pump confirms the device's ability to detect faults with near-perfect accuracy, demonstrating the high sensitivity and effectiveness of the on-device-trained model.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of Interest: The authors declare no conflict of interest.

Author Contributions: All authors contributed to the reported work for conceptualization, methodology, validation, writing-review and editing, and supervision.



تشخیص ناهنجاری مبتنی بر راینش لبه‌ای در ماشین‌های دوار با استفاده از شبکه‌های عصبی مصنوعی

علیرضا مصطفوی، آيسان عليزاده و علی صدیقی*

دانشکده‌ی مهندسی مکانیک، دانشگاه تهران، تهران، ایران.

*نویسنده مسئول (asadighi@ut.ac.ir)

چکیده

در پژوهش حاضر، چارچوبی نوآورانه مبتنی بر فناوری راینش لبه‌ای برای پایش بلادرنگ وضعیت تجهیزات صنعتی ارائه شده است. طراحی سخت‌افزار و ثابت‌افزار به گونه‌ای انجام شده است که تمام وظایف حیاتی از دریافت و پیش‌پردازش داده‌ها تا استخراج ویژگی‌ها و آموزش الگوریتم با وجود محدودیت حافظه و توان پردازشی بر روی گره‌ی لبه‌ای، یعنی میکروکنترلر، ممکن باشد. با دریافت داده‌های ارتعاشی به وسیله‌ی یک شتاب‌سنج سه‌محوره و ذخیره‌سازی داده‌ها برای آموزش در حافظه‌ی فلش میکروکنترلر، یک خودرمزگذار با سه لایه‌ی پنهان برای مدل‌سازی رفتار سالم ماشین بر روی لبه آموزش داده شده و خطای بازسازی داده‌ها برای شناسایی ناهنجاری‌ها استفاده شده است. طبق بررسی‌های انجام‌شده، احتمالاً پژوهش حاضر، اولین پژوهشی است که یک شبکه‌ی عصبی مصنوعی را روی میکروکنترلر آموزش داده و کل فرآیند پایش وضعیت را بر روی لبه پیاده‌سازی کرده است. آزمایش چارچوب پیشنهادی روی یک پمپ گریز از مرکز نشان می‌دهد که سیستم پیشنهادی می‌تواند انواع عیوب پمپ را با دقت بالای ۹۹/۹٪ شناسایی کند.

اطلاعات مقاله

تاریخ دریافت: ۱۴۰۳/۱۰/۰۴

تاریخ اصلاحیه: ۱۴۰۴/۰۳/۱۱

تاریخ پذیرش: ۱۴۰۴/۰۴/۲۹

واژگان کلیدی:

راینش لبه‌ای، آموزش برخط، تشخیص ناهنجاری، خودرمزگذار، ماشین‌آلات دوار.

۱. مقدمه

امروزه، ماشین‌آلات دوار، از جمله: پمپ‌ها، کمپرسورها، توربین‌های گازی، و بادی نقش حیاتی در صنایع ایفا می‌کنند و خرابی غیرمنتظره‌ی آن‌ها می‌تواند منجر به خسارت‌های مالی هنگفت و حتی تلفات جانی شود. لذا، استفاده از پایش وضعیت (CM)^۱ و مدیریت سلامت پیش‌بینانه (PHM)^۲ برای شناسایی عیوب در مراحل اولیه و جلوگیری از وقوع خرابی‌های فاجعه‌بار ضروری است. روش‌های سنتی نگهداری نیازمند بازرسی‌های دوره‌ای هستند، که زمان‌بر، پرهزینه، نیازمند نیروی انسانی زیاد، و تا حدی مبتنی بر قضاوت فردی هستند. در چند دهه‌ی گذشته، پژوهشگران و مهندسان، روش‌های هوشمند پایش وضعیت و نگهداری پیش‌بینانه را به‌منظور کاهش هزینه‌های نگهداری ماشین‌آلات توسعه داده‌اند. همچنین تلاش‌های گسترده و مداومی برای خودکارسازی فرآیند اخیر با استفاده از الگوریتم‌های هوشمند و فناوری‌های پیشرفته برای اندازه‌گیری انجام شده است، تا اطلاعات بلادرنگ در مورد وضعیت دستگاه‌های صنعتی فراهم شود و نواقص احتمالی بدون نیاز به مداخلات مکرر انسانی شناسایی و برطرف شوند. با این حال، موضوع ذکرشده با چالش‌های بسیاری همراه است؛ زیرا از یک سو، هر ماشین مجزا، ویژگی‌های منحصر به خود را دارد و از سوی دیگر، لازم است حجم وسیعی از داده‌ها

جمع‌آوری و با استفاده از منابع محاسباتی گسترده پردازش شوند. روش‌های پایش وضعیت هوشمند به‌طورکلی به دو گروه اصلی تقسیم می‌شوند: (۱) روش‌های مبتنی بر فیزیک، (۲) روش‌های داده‌محور.

(۱) روش‌های مدل‌سازی مبتنی بر فیزیک به‌عنوان راه‌حل اولیه برای شناسایی ناهنجاری‌ها در ماشین‌ها به کار گرفته می‌شدند، که در شرایط خاص عملکرد خوبی از خود نشان می‌دادند.^[۱] با این حال، استفاده از روش‌های مبتنی بر فیزیک نیازمند دانش پیشین گسترده درباره‌ی سیستم و یک مدل دقیق از ماشینی است که باید پایش شود. به وضوح مشخص است که یافتن مدل ریاضی دقیق برای هر ماشین به‌منظور پایش وضعیت آن، کاری پیچیده، چالش‌برانگیز، و گاهی غیرممکن است.^[۲]

(۲) رویکردهای داده‌محور، که از الگوریتم‌های یادگیری ماشین و هوش مصنوعی استفاده می‌کنند، رهیافتی نویدبخش برای مقابله با مسئله‌ی اخیر و غلبه بر محدودیت روش‌های مبتنی بر مدل هستند. روش‌های مذکور با حجم وسیعی از داده‌ها، مدلی برای توصیف وضعیت ماشین ایجاد می‌کنند. حسگرهای متنوع و ارزان‌قیمت بسیاری، از جمله: شتاب‌سنج‌ها، حسگرهای دما، و کرنش‌سنج‌ها می‌توانند داده‌های موردنیاز را برای آموزش الگوریتم‌های

² Prognostics and Health Management¹ Condition Monitoring

یادگیری فراهم کنند. با این حال، مدیریت این حجم وسیع از داده‌ها، چالش‌برانگیز و نیازمند سخت‌افزار گسترده‌ای برای انتقال، ذخیره‌سازی، و پردازش داده‌هاست.

به‌منظور غلبه بر مشکل ذکرشده، مطالعات متعددی به موضوع انتقال داده‌ها به سرورهای ابری پرداخته‌اند، تا از طریق آن‌ها، پردازش داده‌ها و آموزش الگوریتم‌های مناسب برای پایش وضعیت ماشین‌آلات انجام شود. در برخی مطالعات،^۳ و^۴ چارچوب‌های مبتنی بر ابر برای پایش وضعیت سلامت ماشین‌های صنعتی با استفاده از سیگنال‌های ارتعاشی پیشنهاد شده است. وو^۳ و همکاران (۲۰۱۷)،^{۱۳} با بیشتر از ۵۰ حسگر، از جمله مبدل‌های جریان و شتاب‌سنج‌ها، به جمع‌آوری داده‌های لحظه‌ای پمپ‌ها و ماشین‌های سی‌ان‌سی با استفاده از یک میکروکنترلر و مبدل آنالوگ به دیجیتال پرداخته و داده‌های حسگرها را از طریق یک مازول بی‌سیم اکس‌بی^۴ به سرور ابری انتقال داده‌اند. در نهایت، الگوریتم‌های یادگیری ماشین بر روی سرور ابری برای ایجاد یک مدل پیش‌بینی جهت پایش ماشین‌های صنعتی استفاده شده‌اند.

به‌طور کلی، در رویکردهای مبتنی بر رایانش ابری، یک گرهی لبه‌ای مجهز به سیستم داده‌برداری، داده‌های ارتعاشی را از ماشین‌ها جمع‌آوری و آن‌ها را به‌صورت بی‌سیم به یک گرهی می^۵ یا سرور ابری ارسال می‌کند. در نهایت، الگوریتم‌های یادگیری ماشین روی سرور ابری برای شناسایی رفتارهای غیرعادی ماشین استفاده می‌شوند. با این حال، محاسبات ابری با محدودیت‌های عملی مواجه است. انتقال حجم زیادی از داده‌ها به سرور ابری و پردازش آن‌ها می‌تواند مشکلاتی مانند ایجاد گلوگاه را به همراه داشته باشد،^{۱۵} به این معنا که به دلیل زمان پاسخ محدود سرور و پهنای باند شبکه، در انتقال، پردازش، و دریافت داده‌ها تأخیر ایجاد می‌شود. با افزایش تعداد تجهیزات و نیاز به استفاده از حسگرهای متعدد برای پایش مؤثر آن‌ها، وضعیت اخیر تشدید می‌شود. استفاده از سرورهای قدرتمند می‌تواند این مشکل را تا حدودی حل کند؛ اما این امر هزینه‌ی کلی سیستم را افزایش می‌دهد و ممکن است از نظر اقتصادی توجیه‌پذیر نباشد.

رایانش لبه‌ای، یک راهکار اقتصادی و کارآمد برای حل مشکل مذکور ارائه می‌دهد و امکان پردازش داده‌ها و انجام محاسبات ریاضی و آماری را مستقیماً بر روی یک میکروکنترلر در نزدیکی دستگاه تولیدکننده‌ی داده فراهم می‌کند.^{۱۶} رویکرد اخیر به یک تغییر بنیادین در نحوه‌ی پردازش اطلاعات منجر شده و پایش وضعیت تجهیزات نیز از این تحول بی‌بهره نمانده است. با روش رایانش لبه‌ای، داده‌های حسگر مستقیماً در گرهی لبه‌ای پردازش و تحلیل می‌شوند، این امر نیاز به پهنای باند شبکه و منابع سرور را کاهش می‌دهد. در سال‌های اخیر، برخی از پژوهشگران از روش رایانش لبه‌ای برای توسعه‌ی چارچوب‌های پایش وضعیت بلادرنگ استفاده کرده‌اند.^{۵، ۷، ۸} پارک^۶ و همکاران (۲۰۱۸)،^{۱۵} یک شبکه‌ی عصبی بازگشتی را بر روی یک رایانه‌ی شخصی (PC) آموزش داده و سپس آن را به یک رایانه‌ی تک‌برد، رزبری‌پی، منتقل کرده‌اند، تا برای تشخیص عیوب احتمالی یک اجکتور خلأ استفاده شود.

آن‌ها همچنین از یک حسگر شتاب‌سنج پیزوالکتریک در گرهی لبه بهره برده‌اند تا دستگاهی لبه‌محور ایجاد کنند، که توانایی جمع‌آوری، پردازش، و تحلیل داده‌ها را داشته باشد. چیان^۸ و همکاران (۲۰۱۹)،^{۱۷} از یک میکروکنترلر STM32 برای پیاده‌سازی شبکه‌ی عصبی آموزش‌دیده و سیستم داده‌برداری خود استفاده کرده‌اند، تا دستگاهی برای عیب‌یابی بلادرنگ و کنترل دینامیکی موتورهای دی‌سی بدون جاروبک ایجاد کنند. هوانگ^۹ و همکاران (۲۰۲۱)،^{۱۸} از هر دو مفهوم رایانش ابری و لبه‌ای استفاده کرده‌اند. آن‌ها با استفاده از یک گرهی لبه‌ای، داده‌های خام را جمع‌آوری و ویژگی‌هایی را از آن‌ها استخراج کرده‌اند. سپس داده‌های پیش‌پردازش‌شده را به یک سرور ابری انتقال داده‌اند تا یک مدل تشخیص ناهنجاری مبتنی بر دوقلوی دیجیتال^{۱۰} روی ابر ساخته شود. در نهایت، مدل مذکور مجدداً به لایه‌ی لبه‌ای برای پایش مستمر ماشین منتقل شده است. اگرچه با رویکرد اخیر، هزینه‌ی بالای انتقال داده به سرورهای ابری و محاسبات ابری حذف شده است، ولی چارچوب مذکور فاقد تعمیم‌پذیری و قابلیت یادگیری خودکار است.^۹ به این معنا که به دانش قبلی در مورد تجهیزات و مشخصه‌های خرابی آن‌ها به‌منظور آموزش مدل روی رایانه‌ی شخصی یا ابر و سپس انتقال آن به گرهی لبه‌ای نیاز است. هر ماشین، مشخصه‌های منحصر به فردی دارد؛ بنابراین، فرایند اخیر باید برای هر تجهیز که نیاز به پایش دارد، به‌طور مجزا تکرار شود. به عبارت دیگر، سیستم‌های مذکور به‌صورت ترکیبی از ابر و لبه یا رایانه‌ی شخصی و لبه کار می‌کنند و استفاده‌ی آن‌ها از هوش لبه‌ای به اندازه‌ی کافی جامع و فراگیر نیست. بدین ترتیب واضح است که یک الگوریتم پایش وضعیتی باید انتخاب شود که مستقیماً بر روی گرهی لبه آموزش ببیند و بتواند بدون تحمیل هزینه‌ی محاسباتی سنگین و بدون اطلاع از ماهیت خرابی پیش رو، قابلیت یادگیری خودکار را داشته باشد.

انتخاب الگوریتم مناسب، نقش کلیدی در طراحی یک دستگاه مستقل و بلادرنگ برای تشخیص ناهنجاری ماشین‌آلات صنعتی ایفا می‌کند. در شرایط واقعی، هنگام شناسایی رفتار غیرعادی یک ماشین، دسترسی به تمامی مشخصه‌های بالقوه‌ی عیوب آن ممکن نیست. بنابراین، به الگوریتمی با عنوان آشکارسازهای نویی^{۱۱} نیاز است که بتواند رفتار سالم ماشین را یاد بگیرد و الگوهای ناهنجار و عیوب را از آن تشخیص دهد. مدل‌های مذکور می‌توانند منحصراً بر روی داده‌های جمع‌آوری‌شده از ماشین‌آلات در وضعیت نرمال آموزش ببینند و هرگونه ناهنجاری در شرایط کاری را تشخیص دهند. تحلیل مؤلفه‌های اصلی (PCA)^{۱۲}، نقشه‌های خودسازمانده (SOM)^{۱۳}، و خودرمزگذارها^{۱۴} از جمله‌ی الگوریتم‌هایی هستند که توسط پژوهشگران در شرایط مختلف استفاده شده‌اند.^{۱۰} خودرمزگذارها، یکی از بهترین و معروف‌ترین الگوریتم‌های تشخیص ناهنجاری برای ماشین‌آلات دوار به‌شمار می‌روند.^{۱۱، ۱۲، ۱۸}

معصوم و همکاران (۲۰۲۰)،^{۱۱} از یک خودرمزگذار همراه با یک روش جامع مهندسی ویژگی برای شناسایی عیوب در ماشین‌های دوار با استفاده از سیگنال‌های ارتعاشی استفاده کرده و اثربخشی مدل خود را با آزمایش آن روی

⁹ Huang

¹⁰ Digital twin

¹¹ Novelty detectors

¹² Principal Component Analysis

¹³ Self-Organizing Maps

¹⁴ Autoencoder

³ Wu

⁴ Xbee

⁵ Fog node

⁶ Park

⁷ Personal Computer

⁸ Qian

ماشین را به اپراتور ارسال کند. الگوریتم‌های هوش مصنوعی و یادگیری ماشین، همواره نقش کلیدی در دستگاه‌های مبتنی بر اینترنت اشیاء ایفا کرده‌اند. با این حال، آموزش الگوریتم‌های مذکور روی دستگاه‌های اینترنت اشیاء تاکنون چالش‌برانگیز بوده است؛ لذا در پژوهش حاضر، به مشکل مذکور پرداخته شده و با آزمایش دستگاه توسعه‌یافته بر روی یک ماشین صنعتی، یعنی یک پمپ گریز از مرکز، تحقق‌پذیری روش نوآورانه اثبات شده است. به عبارت دیگر، برخلاف بیشتر مطالعاتی که مدل‌های خود را به صورت آفلاین بر روی کامپیوترهای شخصی یا سرورهای ابری آموزش داده‌اند، که فقط برای یک ماشین خاص قابل استفاده هستند، ابزاری توسعه داده شده است که قادر به یادگیری رفتار هر ماشین دوار به صورت برخط و شناسایی ناهنجاری‌های قریب‌الوقوع آن است.

ادامه‌ی نوشتار حاضر بدین ترتیب سازماندهی شده است: در بخش ۲، به مهندسی ویژگی و الگوریتم تشخیص ناهنجاری پرداخته شده است. بر روی سیستم آزمایش و طراحی معماری الگوریتم در بخش ۳ بحث شده و در بخش ۴، معماری و مشخصات دستگاه ارائه شده است. توضیح روش‌های رایانش لبه‌ای و فرایندهای آموزش آنلاین در بخش ۵، و در نهایت، جمع‌بندی نتایج پژوهش حاضر در بخش ۶، ارائه شده است.

۲. روش پیشنهادی

۱.۲. پیش‌پردازش داده‌ها و مهندسی ویژگی

مهندسی ویژگی، شامل استخراج اطلاعات مفید از داده‌های خام است و به عنوان سنگ بنای شناسایی موفق ناهنجاری‌ها در نظر گرفته می‌شود و برای کاهش ابعاد داده‌ها و حذف مقادیر زائد در مجموعه‌ی داده‌ها به کار می‌رود. بنابراین، پژوهشگران در تمامی نوشتارهای مرتبط با تشخیص عیوب، توجه ویژه‌ای به این موضوع داشته و از روش‌های مختلف در مطالعات موردی خود استفاده کرده‌اند. در سال‌های اخیر، روش‌های یادگیری عمیق، مانند: شبکه‌های عصبی پیچشی (CNN)^{۱۹}، خودرم‌گذارها، و شبکه‌های عصبی بازگشتی (RNN)^{۲۰} برای استخراج ویژگی از طریق فرآیند یادگیری استفاده شده‌اند.^{۵، ۱۲، ۱۸} با این حال، برخی پژوهشگران ترجیح داده‌اند که ویژگی‌های مؤثر را به صورت دستی انتخاب کنند.^{۳، ۷، ۸، ۱۱}

با توجه به محدودیت‌های مرتبط با توان پردازشی و حافظه در میکروکنترلرها، که اغلب به چند کیلوبایت محدود است و با هدف انجام کل فرایند روی لبه، ذخیره‌سازی داده‌های خام روی لبه یا ساخت مدل یادگیری عمیق برای استخراج ویژگی‌ها عملی نیست. در نتیجه، برجسته‌ترین ویژگی‌ها به گونه‌ای طراحی می‌شوند که ابعاد داده‌های نمونه کاهش یابند و مؤثرترین اطلاعات از داده‌های ارتعاشی خام ماشین‌های دوار استخراج شوند.

ویژگی‌های مرتبط با تشخیص ناهنجاری، معمولاً در سه گروه اصلی طبقه‌بندی می‌شوند: حوزه‌ی زمان، حوزه‌ی بسامد، و حوزه‌ی زمان-بسامد. استفاده از سطح مجذور میانگین مربعات سیگنال (RMS)^{۲۱}، ضریب اوج^{۲۲}، و نسبت بیشینه‌ی مقدار سیگنال به مقدار مؤثر آن، رایج‌ترین روش‌های استخراج ویژگی‌های

یک موتور دی‌سی بدون جاروبک اثبات کرده و دریافته‌اند که روش پیشنهادی برای یادگیری برخط و کاربردهای بلادرنگ مناسب است. احمد و همکاران (۲۰۲۰)^{۱۲} از یک خودرم‌گذار همراه با یک شبکه‌ی عصبی حافظه‌ی کوتاه‌مدت بلند (LSTM)^{۱۵} برای استخراج ویژگی از سیگنال‌های ارتعاشی خام و تشخیص ناهنجاری‌ها در ماشین‌های دوار استفاده کرده‌اند.

در رابطه با میکروکنترلرها و کامپیوترهای تعبیه‌شده، محدودیت‌هایی از جمله حافظه‌ی محدود و توان پردازشی کم مطرح است. به همین دلیل نمی‌توان الگوریتم‌های هوش مصنوعی و یادگیری ماشین را به سادگی و مانند آموزش آن‌ها روی لپ‌تاپ یا سرورهای قدرتمند با CPU (واحد پردازش مرکزی) یا GPU (واحد پردازش گرافیکی) سطح بالا و حافظه‌ی تقریباً نامحدود آموزش داد. حتی تولیدکنندگان میکروکنترلرها پیشنهاد می‌کنند که شبکه‌های عصبی پس از آموزش روی رایانه‌های شخصی به میکروکنترلرها منتقل شوند.^{۱۳} به علاوه، کتابخانه‌هایی برای توسعه‌ی شبکه‌های عصبی مصنوعی روی میکروکنترلرها موجود هستند، مانند: Tensorflow Lite for Microcontrollers توسعه‌یافته توسط گوگل،^{۱۴} CMSIS NN از Arm،^{۱۵} X-CUBE-AI از ST،^{۱۶} و TVM از Apache.^{۱۷} با این حال، در تمامی کتابخانه‌های مذکور فرض شده است که شبکه پس از آموزش روی کامپیوترهای قدرتمند به صورت ماتریس‌هایی از اعداد به میکروکنترلرها منتقل شده‌اند، به این معنی که آن‌ها از آموزش روی دستگاه پشتیبانی نمی‌کنند. از سوی دیگر، دستیابی به یک دستگاه مستقل برای تشخیص ناهنجاری در ماشین‌آلات دوار مستلزم آموزش الگوریتم بر روی لبه است. در همین راستا، برخی پژوهشگران تلاش کرده‌اند راه‌حلی برای این موضوع ارائه دهند. به عنوان مثال، رن^{۱۶} و همکاران (۲۰۲۱)^{۱۸} سیستمی نوآورانه به نام TinyOL (یادگیری ماشین کوچک با یادگیری آنلاین) پیشنهاد کرده‌اند. مدل آن‌ها شامل یک خودرم‌گذار از پیش‌آموزش‌دیده است، که لایه‌ای به نام TinyOL به لایه‌ی نهایی خودرم‌گذار متصل شده است؛ که مسئولیت آموزش پس از انتقال آخرین لایه‌ی شبکه را در حالت آنلاین بر عهده دارد. در واقع، آن‌ها مدل را پس از انتقال به میکروکنترلر فقط تنظیم دقیق^{۱۷} می‌کنند و کل مدل را بر روی لبه آموزش نمی‌دهند. آن‌ها با استفاده از بُرد Arduino Nano 33 BLE که دارای پردازنده‌ی Cortex™-M4 با بسامد ۶۴ مگاهرتز و ۲۵۶ کیلوبایت حافظه‌ی SRAM است، مدل خود را برای تشخیص ناهنجاری یک فن کامپیوتر آزمایش کرده و با ۱۰۰۰ داده‌ی آموزشی جمع‌آوری‌شده از فن سالم توانسته‌اند به نمره‌ی F1^{۱۸} برابر با ۱ دست یابند.

در پژوهش حاضر، یک افزار نوآورانه‌ی مستقل مبتنی بر رایانش لبه‌ای برای تشخیص ناهنجاری ماشین‌های دوار توسعه داده شده است؛ که با اتصال به یک ماشین دوار می‌تواند سیگنال‌های ارتعاشی خام را از طریق شتاب‌سنج دریافت کند، سپس به پیش‌پردازش داده‌های خام پردازد، و ویژگی‌های مؤثر را از آن استخراج کند؛ و در نهایت، با ذخیره‌سازی داده‌های آموزش بر روی لبه، یک خودرم‌گذار را کاملاً روی میکروکنترلری با فقط ۱ مگابایت حافظه‌ی رم به منظور تشخیص ناهنجاری در رفتار ماشین به صورت بلادرنگ آموزش دهد. این دستگاه همچنین به یک مازول بی‌سیم مجهز است تا وضعیت لحظه‌ای

²⁰ Recurrent Neural Network

²¹ Root Mean Square

²² Crest factor

¹⁵ Long Short-Term Memory

¹⁶ Ren

¹⁷ Fine-tune

¹⁸ F1-Score

¹⁹ Convolutional Neural Network

جدول ۱. ویژگی‌های حوزه‌ی زمان.

Table 1. Time domain features.

Formula	Description	Formula	Description
$X_{rms} = \left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right)^{\frac{1}{2}}$	Root Mean Squared of Signal (RMS)	$X_{cf} = \frac{\max(x_i)}{\left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right)^{\frac{1}{2}}}$	Crest factor
$X_{sra} = \left(\frac{1}{N} \sum_{i=1}^N \sqrt{ x_i } \right)^2$	Square root of the amplitude	$X_{gf} = \frac{\max(x_i)}{\frac{1}{N} \sum_{i=1}^N x_i }$	Impulse Factor
$X_{kv} = \frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \bar{x}}{\sigma} \right)^4$	Kurtosis value	$X_{mf} = \frac{\max(x_i)}{\left(\frac{1}{N} \sum_{i=1}^N \sqrt{ x_i } \right)^2}$	Margin Factor
$X_{sv} = \frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \bar{x}}{\sigma} \right)^3$	Skewness Value	$X_{sf} = \frac{\left(\frac{1}{N} \sum_{i=1}^N x_i^3 \right)^{\frac{1}{2}}}{\frac{1}{N} \sum_{i=1}^N x_i }$	Shape Factor
$X_{ppv} = \max(x_i) - \min(x_i)$	Peak-peak value	$X_{kf} = \frac{\frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \bar{x}}{\sigma} \right)^4}{\left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right)^2}$	Kurtosis Factor

جدول ۲. ویژگی‌های حوزه‌ی بسامد.

Table 2. Frequency domain features.

Formula	Description
$X_{fc} = \frac{\sum_{i=0}^{N-1} f_i s(f_i)}{\sum_{i=0}^{N-1} s(f_i)}$	Frequency center
$X_{rmsf} = \left(\frac{\sum_{i=0}^{N-1} f_i^2 s(f_i)}{\sum_{i=0}^{N-1} s(f_i)} \right)^{\frac{1}{2}}$	RMS frequency
$X_{rmsf} = \left(\frac{\sum_{i=0}^{N-1} (f_i - X_{fc})^2 s(f_i)}{\sum_{i=0}^{N-1} s(f_i)} \right)^{\frac{1}{2}}$	Root variance frequency

انرژی گره‌های لایه‌ی آخر، که شامل ۱۶ گره است، به عنوان ویژگی‌های حوزه‌ی زمان - بسامد انتخاب می‌شوند. هر نمونه‌ی ارتعاشی، که از طریق شتاب‌سنج از هر محور به‌دست می‌آید، شامل ۲۰۴۸ نقطه‌ی نمونه است. پس از ذخیره‌ی داده‌های خام ارتعاشی سه محور در سه آرایه‌ی جداگانه، بهره‌ی دی‌سی هر محور با کسر میانگین آرایه از هر نقطه‌ی نمونه حذف شده است. سپس، ۲۹ ویژگی از هر محور استخراج و در یک آرایه با طول ۸۷ ذخیره شده است.

۲.۲. الگوریتم شناسایی ناهنجاری

از آنجایی که هدف از پژوهش حاضر ایجاد یک دستگاه مستقل برای عیب‌یابی بلادرنگ ماشین‌های دوار است، نمی‌توان از روش‌های رایج طبقه‌بندی استفاده

حوزه‌ی زمان هستند.^[۱۹] مقدار کشیدگی^{۲۳} و مقدار پیک، نیز از دیگر ویژگی‌های مهم در حوزه‌ی زمان هستند، که در نوشتار ویلیامز و همکاران (۲۰۰۱،^[۲۰] برای عیب‌یابی در یاتاقان‌های غلتشی و ساچمه‌ای به‌کار رفته‌اند. در کنار ویژگی‌های اخیر، برخی پارامترهای بی‌بعد، مانند: ضریب شکل، ضریب ضربه، ضریب کشیدگی، و ضریب حاشیه نیز در نوشتار ایکسیا^{۲۴} و همکاران (۲۰۱۲،^[۲۱] برای پیش‌بینی عیوب در یاتاقان استفاده شده‌اند. در جدول ۱، تمام ویژگی‌های مرتبط با حوزه‌ی زمان، که در مطالعه‌ی حاضر انتخاب شده‌اند، ارائه شده‌اند؛ که در آن‌ها، X داده‌های خام ارتعاشی، $\bar{x}$ میانگین داده‌ها، σ انحراف معیار X ، و N تعداد کل نقاط نمونه در یک نمونه‌ی ارتعاشی هستند.

استفاده از تبدیل فوریه‌ی سریع (FFT)^{۲۵}، رایج‌ترین روش برای استخراج اطلاعات مفید از سیگنال خام در حوزه‌ی بسامدی است. در جدول ۲، سه ویژگی شناخته‌شده در حوزه‌ی بسامدی ارائه شده است، که براساس FFT سیگنال‌های ارتعاشی حوزه‌ی زمان محاسبه شده‌اند.^{[۱۱] و [۲۱]} در فرمول‌های اخیر، $s(f)$ تبدیل فوریه‌ی سریع نمونه، و f_i بسامدی است که $s(f_i)$ در آن محاسبه می‌شود.

به‌دلیل کاستی‌های موجود در ویژگی‌های حوزه‌های زمان و بسامد در آنالیز سیگنال‌های غیرایستا، ویژگی‌های حوزه‌ی زمان - بسامد، به‌عنوان بهترین ویژگی‌ها برای سیگنال‌های غیرایستا در نظر گرفته شده‌اند.^[۲۲] تبدیل فوریه‌ی زمان - کوتاه (STFT)^{۲۶}، تبدیل موجک پیوسته (CWT)^{۲۷}، تبدیل موجک گسسته (DWC)^{۲۸}، و تجزیه‌ی بسته‌های موجک (WPD)^{۲۹} در گروه مذکور جای می‌گیرند. ویژگی‌های حوزه‌ی زمان - بسامد برای عیب‌یابی در ماشین‌های دوار به‌طور جامع در نوشتار معصوم و همکاران (۲۰۲۰،^[۱۱] بررسی شده‌اند. در پژوهش حاضر، از WPD به‌عنوان بهترین و مؤثرترین روش برای استخراج اطلاعات در این حوزه استفاده شده است. با استفاده از WPD، سیگنال به بخش‌های بسامدهای پایین و بالا، سطح به سطح تا سطح چهارم تجزیه می‌شود.

²⁷ Continuous Wavelet Transform

²⁸ Discrete Wavelet Transform

²⁹ Wavelet Packet Decomposition

²³ Kurtosis value

²⁴ Xia

²⁵ Fast Fourier Transform

²⁶ Short-Time Fourier Transform

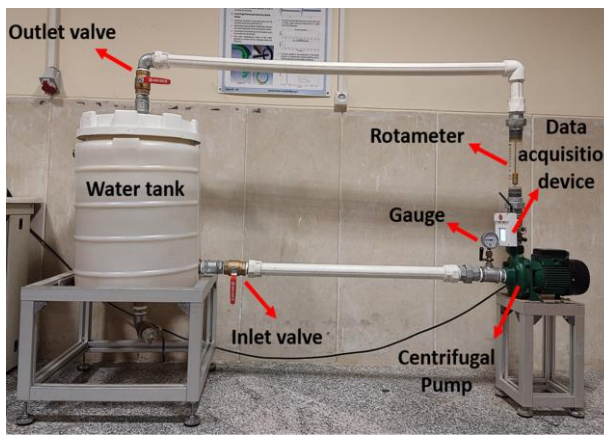


Figure 2. Experimental Setup.

شکل ۲. نمای کلی از سیستم آزمایش.

در پژوهش حاضر، یک شبکه‌ی خودرمزگذار عمیق با چندین لایه‌ی مخفی و الگوریتم پس‌انتشار برای آموزش آن با استفاده از زبان برنامه‌نویسی C طراحی شده است. کد مربوط به تمامی مراحل استخراج ویژگی‌ها و آموزش مدل با استفاده از CodeBlocks، که یک محیط متن‌باز و کامپایلر C است، روی یک لپ‌تاپ با پردازنده‌ی اینتل تک‌هسته‌ای با سرعت ۱/۸ گیگاهرتز نوشته و اجرا شده است.^[۲۳] در بخش بعد، جزئیات کاملی از معماری پیشنهادی خودرمزگذار ارائه شده است.

۳. طراحی آفلاین معماری خودرمزگذار

۱.۳. سیستم آزمایش

برای طراحی و تعیین پارامترهای خودرمزگذار، از جمله تعداد لایه‌های پنهان، تعداد نورون‌ها در هر لایه، و نرخ یادگیری، مقرر شد داده‌هایی تحت شرایط سالم و معیوب از یک ماشین دوار جمع‌آوری شوند. برای این کار، از یک سیستم آزمایش هیدرولیکی استفاده شده است، که در آن یک پمپ گریز از مرکز می‌تواند با استفاده از شیرهای ورودی و خروجی در شرایط عملیاتی مختلف تنظیم شود. در این حین، نرخ جریان و فشار سیال با استفاده از روماتر و دو گیج فشار پایش می‌شوند. نمای کلی سیستم آزمایش در شکل ۲ مشاهده می‌شود.

توان ورودی پمپ گریز از مرکز توسط یک موتور القایی تک‌فاز با دو قطب، توان یک اسب بخار، و ولتاژ نامی ۲۲۰ ولت تأمین شده است. یک دستگاه داده‌برداری نیز با استفاده از دو آهن‌ربا به پمپ متصل شده است؛ که شامل محفظه، بُرد توسعه‌ی ST، باتری، حسگر شتاب‌سنج، و ماژول بی‌سیم اکس‌بی بوده است. بُرد توسعه از نوع STM32F429 (تولید شرکت STMicroelectronics) با بسامد کاری ۱۶۸ مگاهرتز، سیگنال شتاب‌سنج را با نرخ نمونه‌برداری ۱۶۰۰ هرتز دریافت و با استفاده از ماژول اکس‌بی به یک سرور برای ذخیره‌سازی ارسال می‌کند. یک فیلتر پایین‌گذر با بسامد گوشه‌ی ۸۰۰ هرتز برای جلوگیری از پدیده‌ی الایسینگ^{۳۴} استفاده شده است. علاوه بر این، دستگاه یک صفحه‌ی نمایش ال‌سی‌دی دارد، که تعداد سری‌های سیگنال شتاب ارسال‌شده و زمان تغییر نقاط عملیاتی پمپ را نشان می‌دهد.

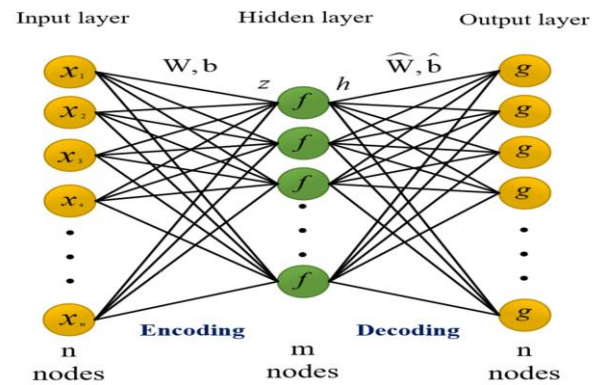


Figure 1. an autoencoder with one hidden layer.

شکل ۱. خودرمزگذار با یک لایه‌ی پنهان.

کرد؛ زیرا آموزش آن‌ها نیاز به داده‌های برچسب‌گذاری‌شده از هر دو حالت سالم و معیوب ماشین دارد، که در بیشتر موارد در دسترس نیستند. در این شرایط، الگوریتم‌های شناسایی ناهنجاری بهترین گزینه هستند. در مطالعه‌ی حاضر، از یکی از پرکاربردترین الگوریتم‌های شناسایی ناهنجاری، یعنی خودرمزگذارها استفاده شده است. خودرمزگذارها شبکه‌های عصبی مصنوعی پیش‌خور^{۳۰} و کاملاً متصل^{۳۱} هستند، که می‌توانند آموزش ببینند تا ورودی‌های بدون برچسب را بازسازی کنند و شامل دو بخش هستند: رمزگذار^{۳۲} و رمزگشا^{۳۳}. بخش رمزگذار شبکه تلاش می‌کند در عین حفظ اطلاعات لایه‌ی ورودی، ابعاد آن را از طریق تعدادی لایه‌ی دلخواه کاهش دهد. بخش رمزگشا آموزش می‌بیند تا داده‌های ورودی را از فضای ویژگی (که معمولاً لایه‌ی مرکزی در خودرمزگذار است) در لایه‌ی خروجی بازتولید کند. در شکل ۱، خودرمزگذاری ساده با یک لایه‌ی پنهان مشاهده می‌شود، که در آن $x_1, x_2, x_3, \dots, x_n$ بردار ورودی، f و g توابع فعال‌سازی هستند، که معمولاً توابع سیگموئید $\sigma(x) = 1 / (1 + e^{-x})$ یا تانژانت هایپربولیک $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ هستند.

فضای ویژگی‌ها (h) و خروجی‌های بازسازی‌شده (y) را می‌توان به صورت روابط ۱ و ۲ تعریف کرد:

$$h_i = f(W_{ij}x_j + b_j) \quad (1)$$

$$y_j = g(\hat{W}_{ji}h_i + \hat{b}_i) \quad (2)$$

اگر خودرمزگذار با داده‌های مربوط به حالت سالم ماشین آموزش ببیند، می‌تواند دقیقاً داده‌های سالم را بازتولید کند. اما وقتی داده‌ی معیوب به آن داده شود، قادر به بازسازی آن نخواهد بود. بنابراین، میانگین مربعات خطا (MSE)^{۳۴} بین ورودی و خروجی خودرمزگذار می‌تواند به‌عنوان یک شاخص برای تمایز بین حالت سالم و معیوب ماشین استفاده شود. در نتیجه، ماتریس‌های وزن و بردارهای بایاس در طی فرآیند یادگیری با استفاده از تابع هزینه‌ی MSE و الگوریتم پس‌انتشار^{۳۵} به نحوی تنظیم می‌شوند تا داده‌های سالم بازسازی شوند. این فرآیند به‌صورت رابطه‌ی ۳ انجام می‌شود:

$$L(W, \hat{W}, b, \hat{b}) = \arg \min_{w, \hat{w}, b, \hat{b}} \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2 \quad (3)$$

³⁴ Mean Squared Error

³⁵ Back propagation

³⁶ Aliasing

³⁰ Feed-forward

³¹ Fully connected

³² Encoder

³³ Decoder

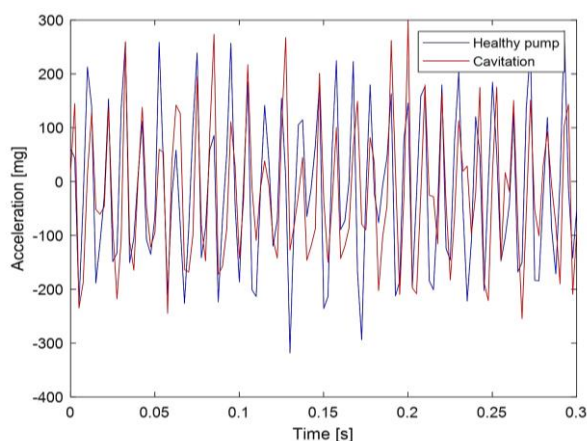


Figure 5. The vibration signal of the pump under healthy and cavitation conditions.

شکل ۵. سیگنال‌های ارتعاشی پمپ تحت شرایط سالم و کاویتاسیون.

همان‌طور که مشخص است، سیگنال‌های ارتعاشی خام برای شرایط سالم و کاویتاسیون، حتی در دامنه، تفاوت‌های قابل توجهی ندارند و تفکیک بین دو سیگنال مذکور خام، کار آسانی نیست. اما همان‌طور که در شکل ۶ مشاهده می‌شود، کاویتاسیون با استفاده از روش‌های مهندسی، قابل شناسایی است.

شکل‌های ۶ (a) و ۶ (b) به ترتیب ویژگی‌های حوزه‌های زمان و بسامد را نشان می‌دهند. محور X نمودارهای مذکور، مخفف نام ویژگی‌های معرفی‌شده در جدول‌های ۱ و ۲ است. در شکل ۶ (c)، انرژی گره‌های لایه‌ی آخر از تجزیه‌ی بسته‌های موجک مشاهده می‌شود. در حالی که ویژگی‌های حوزه‌های زمان و بسامد، تفاوت معنی‌داری بین دو حالت نشان نمی‌دهند، ویژگی‌های حوزه‌ی زمان-بسامد در شکل ۶ (c) به وضوح بین حالت‌های سالم و معیوب تمایز قائل می‌شوند، یعنی که تحلیل موجک در تشخیص حالت‌های معیوب و سالم یک ماشین دوار براساس سیگنال شتاب آن مؤثر است.

با توجه به اینکه تعداد بهینه‌ی لایه‌های پنهان در شبکه‌های خودرمزگذار عمیق به نوع مسئله وابسته است، برای تعیین تعداد بهینه در مسئله‌ی حاضر، ۵ مدل خودرمزگذار با تعداد مختلف از لایه‌های مخفی (با ۱ تا ۹ لایه)، طراحی و آموزش داده شده‌اند. فرآیند آموزش با استفاده از ۴۰۰ نمونه‌ی داده‌ی سالم به‌عنوان داده‌ی آموزشی انجام و عملکرد مدل‌ها با استفاده از ۷۰۰ نمونه‌ی معیوب و ۶۰۰ نمونه‌ی سالم از پیش دیده‌نشده ارزیابی شده است. اندازه‌ی لایه‌های ورودی و خروجی در تمامی شبکه‌ها برابر با تعداد کل ویژگی‌ها (یعنی ۸۷) و اندازه‌ی هر لایه‌ی پنهان با آزمون و خطا تعیین شده است، تا بهترین عملکرد ممکن به‌دست آید.

پس از آموزش همه‌ی مدل‌ها، خطای میانگین مربعات داده‌های آموزشی برای هر شبکه به‌طور مجزا نرمال‌سازی و سطح سه-سیگما به عنوان آستانه‌ی ناهنجاری انتخاب شده است. به این معنا که وقتی یک نمونه توسط شبکه به‌عنوان نمونه‌ی سالم طبقه‌بندی می‌شود، با احتمال ۹۹/۷٪ سالم است. علاوه بر این، ۴ معیار استاندارد برای ارزیابی شبکه‌های عصبی، از جمله: دقت، بازیابی، درستی، و امتیاز F1 برای ارزیابی عملکرد مدل‌ها استفاده شده است؛ که با شمارش تعداد موارد: مثبت صحیح (TP)^{۳۸}، منفی صحیح (TN)^{۳۹}، مثبت



Figure 3. Vane tip fault.

شکل ۳. عیب پریدگی نوک پره.

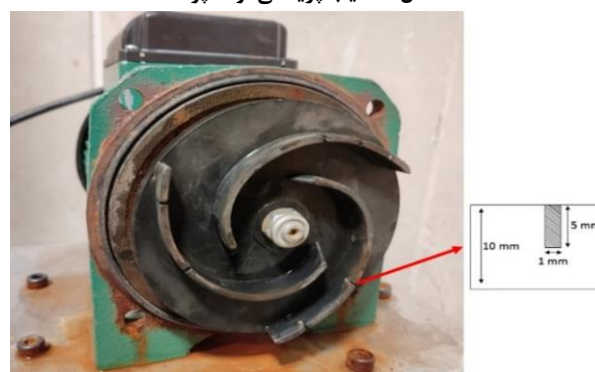


Figure 4. Impeller crack fault.

شکل ۴. عیب ترک شبیه‌سازی‌شده بر روی پره.

برای شبیه‌سازی شرایط کاری معیوب پمپ، سه عیب رایج در پمپ‌های گریز از مرکز شبیه‌سازی شده است:

۱. کاویتاسیون. کاویتاسیون زمانی رخ می‌دهد که فشار مایع اطراف پروانه به زیر فشار بخار مایع کاهش یابد. در نتیجه، مایع تبخیر و به حباب تبدیل می‌شود. زمانی که حباب‌ها به ناحیه‌ای با فشار بالاتر منتقل شوند، منفجر می‌شوند و موج‌های شوک قوی تولید می‌کنند. برخورد این موج‌ها به پروانه باعث فرسایش تدریجی آن می‌شود. این عیب با بازکردن شیر خروجی پمپ و بستن همزمان شیر ورودی آن شبیه‌سازی شده است.

۲. پریدگی نوک پره^{۳۷}. این عیب به دلیل کاویتاسیون، خوردگی، و فرسایش اتفاق می‌افتد. این خرابی با بریدن نوک پره، مطابق شکل ۳ شبیه‌سازی شده است.

۳. ترک روی پره. عوامل مشابهی که در پریدگی نوک پره مؤثر هستند، می‌توانند باعث ایجاد ترک در پره شوند. در شکل ۴، نحوه‌ی شبیه‌سازی این عیب در پروانه‌ی پمپ مشاهده می‌شود.

۲.۳ آموزش آفلاین

در مجموع، ۱۰۰۰ آزمون در شرایط سالم و ۲۸۰۰ آزمون در شرایط معیوب انجام شده است. تمام نمونه‌های معیوب در ۵ نقطه‌ی عملیاتی پمپ، شامل: ۱۲۰، ۱۳۰، ۱۴۰، ۱۵۰، و ۱۶۰ لیتر در دقیقه برای انواع خرابی جمع‌آوری شده‌اند. آزمون‌های مربوط به شرایط سالم در ۱۲ نقطه‌ی عملیاتی، از ۹۰ تا ۲۱۰ لیتر در دقیقه انجام شده‌اند. در شکل ۵، سیگنال‌های شتاب پمپ در هر دو حالت سالم و کاویتاسیون مشاهده می‌شوند.

³⁹ True Negative

³⁷ Vane tip fault

³⁸ True Positive

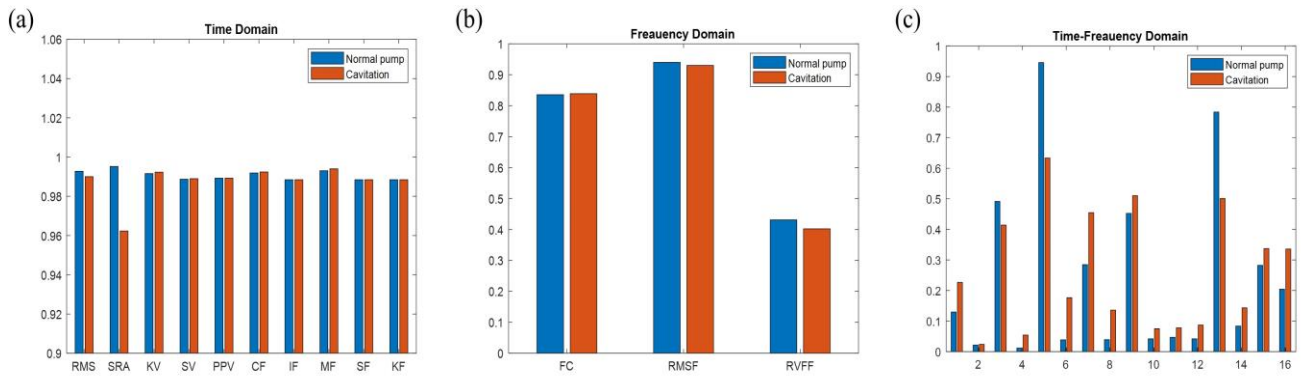


Figure 6. (a) Time domain, (b) Frequency domain, (c) Time-frequency domain extracted features of the pump under healthy and cavitation conditions.

شکل ۶. ویژگی‌های استخراج‌شده از پمپ تحت شرایط سالم و کاویتاسیون در: (a) حوزه‌ی زمان، (b) حوزه‌ی بسامد، (c) حوزه‌ی زمان-بسامد.

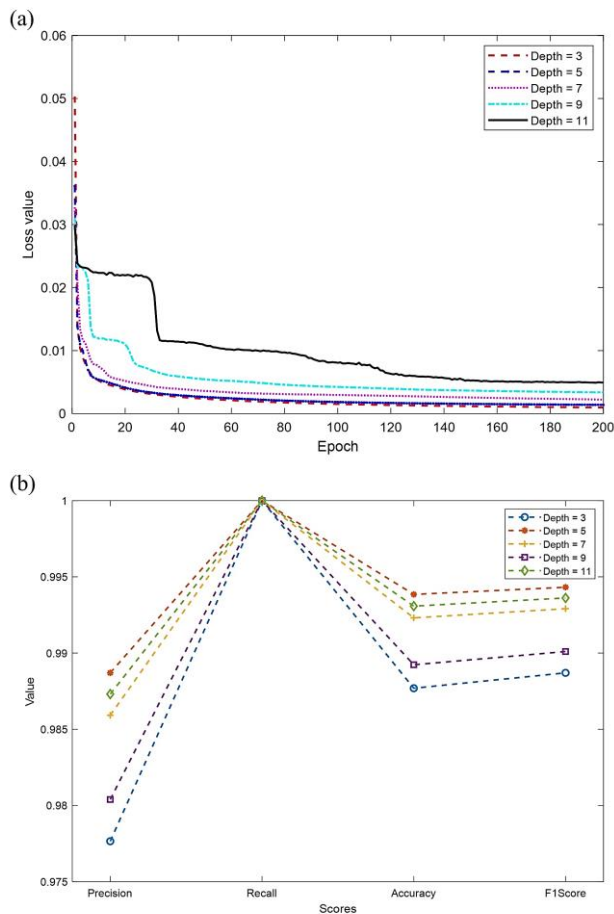


Figure 7. (a) The reconstruction losses of five autoencoders with different depths, (b) Performance of the networks with different depths.

شکل ۷. (a) خطای بازسازی خودرمزگذارها به ازاء عمق‌های مختلف، (b) عملکرد شبکه به ازاء عمق‌های مختلف.

درخصوص انتخاب نرخ یادگیری مناسب، مدل انتخابی با ۵ نرخ یادگیری مختلف و با همان تعداد داده‌های آموزش و آزمون، آموزش داده شده است (شکل ۸). علی‌رغم اینکه شبکه با تمامی نرخ‌های یادگیری به جز مقدار ۰/۰۲۵ به‌طور پایدار آموزش دیده و خطاها به مقدار تقریباً یکسانی همگرا شده‌اند، مدل آموزش دیده با نرخ یادگیری ۰/۰۰۱ از نظر ۴ معیار، عملکرد بهتری نسبت به

جدول ۳. معماری شبکه‌ی خودرمزگذار پیشنهادی.

Table 3. The architecture of the proposed autoencoder.

Layer	No. of Neurons	Activation Function
Input	87	Linear
1st hidden Layer	40	Hyperbolic tangent
2nd hidden layer	25	Sigmoid
3rd hidden layer	40	Hyperbolic tangent
Output	87	Sigmoid

کاذب (FP)^{۴۰}، و منفی کاذب (FN)^{۴۱} با استفاده از معادله‌های ۴ الی ۷ محاسبه شده‌اند.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (6)$$

$$F1-Score = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (7)$$

در شکل ۷ (a)، میزان خطای بازسازی مدل‌ها با افزایش تعداد ایپاک و در شکل ۷ (b)، عملکرد شبکه‌ها براساس ۴ معیار ارزیابی اشاره‌شده مشاهده می‌شوند. مطابق شکل ۷ (a)، فرآیند آموزش شبکه‌هایی با عمق ۳ تا ۷ نسبتاً پایدار بوده و خطاهای آن‌ها به مقادیر پایین و قابل قبولی همگرا شده است. با این حال، مطابق شکل ۷ (b)، شبکه با عمق ۵ در تمامی معیارها عملکرد بهتری نسبت به سایر مدل‌ها داشته است. بنابراین، شبکه با سه لایه‌ی مخفی (عمق ۵) به‌عنوان مدل بهینه برای ادامه‌ی پژوهش انتخاب شده است. معماری شبکه‌ی پیشنهادی در جدول ۳ ارائه شده است. تعداد نورون‌های لایه‌ی ورودی و دو لایه‌ی مخفی بعدی به ترتیب ۸۷، ۴۰، و ۲۵ تعیین شده است.

⁴¹ False Negative

⁴⁰ False Positive

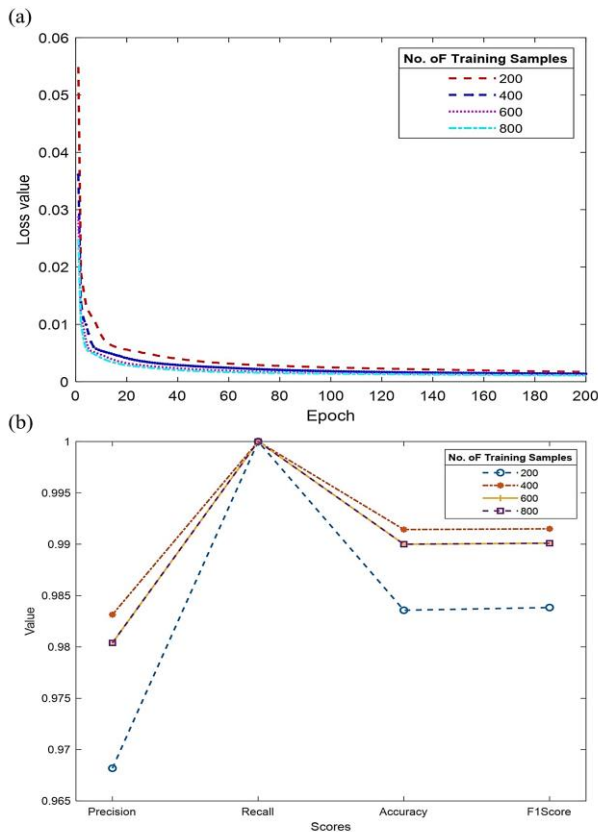


Figure 9. (a) The reconstruction loss, and (b) The performance of the autoencoder for four different training dataset size.

شکل ۹. (a) خطای بازسازی خودرمزگذار به ازاء ابعاد مختلف از دیتاست آموزش، (b) عملکرد خودرمزگذار به ازاء ابعاد مختلف از دیتاست آموزش.

حافظه‌ی محدود و در حدود چند کیلوبایت تا ۱ مگابایت هستند. بنابراین لازم است مدل، الگوریتم آموزشی، و فرآیند استخراج ویژگی تا حد ممکن بهینه شوند؛ به طوری که قابل پیاده‌سازی بر روی میکروکنترلرهای کوچک باشند. از سوی دیگر، امکان ذخیره‌سازی تمام داده‌ها برای آموزش شبکه روی لبه وجود ندارد. به همین دلیل، داده‌ها به صورت دسته‌های کوچک و مرحله‌به‌مرحله به لبه ارسال می‌شوند و شبکه به صورت ترتیبی آموزش می‌بیند. این روش نیاز به حافظه را کاهش می‌دهد و آن را با محدودیت‌های میکروکنترلرها سازگار می‌کند. به عبارت دیگر، شبکه، ابتدا با ذخیره‌ی یک دسته داده در حافظه آموزش داده می‌شود؛ سپس دسته‌ی بعدی در همان فضای حافظه ذخیره می‌شود و فرآیند آموزش ادامه می‌یابد. برای بررسی رویکرد نوآورانه و ترتیبی مذکور، خودرمزگذار با استفاده از ۴۰۰ داده‌ی سالم، در چهار دسته‌ی ۱۰۰ تایی روی یک کامپیوتر آموزش داده شده است. همچنین، در هر دسته، ۱۰٪ از داده‌ها به عنوان مجموعه‌ی داده‌ی راستی‌آزمایی انتخاب شده‌اند. در شکل ۱۰، مقادیر تابع هزینه بر روی مجموعه‌ی داده‌های آموزش و راستی‌آزمایی برای چهار دسته‌ی مذکور مشاهده می‌شود. از طریق آزمون و خطا مشخص شده است که برای آموزش دسته‌ی اول، ۱۵۰ اپیاک نیاز است و برای سه دسته‌ی بعدی، هر دسته با ۵۰ اپیاک به نتیجه‌ی مطلوب می‌رسد. بنابراین، کل فرآیند آموزش در مجموع طی ۳۰۰ اپیاک تکمیل می‌شود. علاوه بر این، از آنجا که مقدار تابع هزینه روی داده‌های راستی‌آزمایی به موازات مقدار آن بر روی مجموعه‌ی داده‌ی آموزش کاهش یافته است، شبکه دچار بیش‌برازش نشده است.

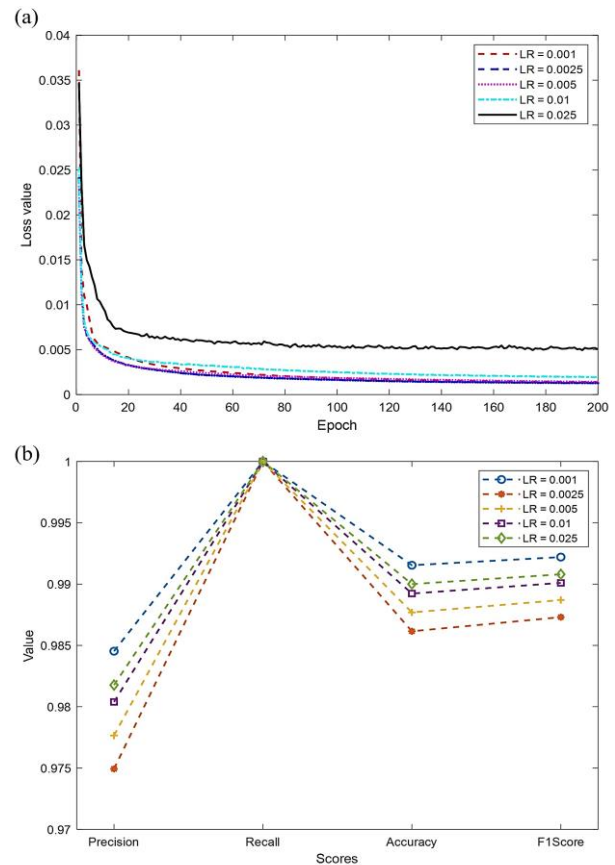


Figure 8. (a) The reconstruction loss, and (b) The performance of the autoencoder with three hidden layers and five different learning.

شکل ۸. (a) خطای بازسازی خودرمزگذار به ازاء نرخ‌های یادگیری مختلف، (b) عملکرد خودرمزگذار به ازاء نرخ‌های یادگیری مختلف.

سایر نرخ‌ها داشته است. بنابراین، در ادامه‌ی نوشتار حاضر، از نرخ یادگیری ۰/۰۰۱ استفاده شده است.

در نهایت، به منظور تعیین تعداد داده‌ی مناسب برای آموزش، مدل با استفاده از ۴ مجموعه‌ی داده با اندازه‌های متفاوت و نرخ یادگیری ۰/۰۰۱ آموزش داده شده است (شکل ۹). با در نظر گرفتن خطای بازسازی داده‌های سالم، مطابق شکل ۹ (a)، مدل در تمامی حالت‌ها عملکرد خوبی داشته و اختلافات جزئی مشاهده شده است. با این حال، بر اساس شکل ۹ (b) به نظر می‌رسد که فقط ۲۰۰ داده‌ی سالم برای آموزش تمامی پارامترهای شبکه کافی نیست و با بیش از ۴۰۰ داده‌ی آموزش، احتمالاً مدل دچار بیش‌برازش می‌شود، زیرا مقادیر چهار معیار با اندازه‌های ۲۰۰، ۶۰۰ و ۸۰۰ کاهش یافته‌اند. بنابراین، اندازه‌ی مجموعه‌ی داده برابر با ۴۰۰ برای آموزش مدل مناسب است. علاوه بر این، در شکل‌های ۷ (b)، ۸ (b)، و ۹ (b) مقادیر بازیابی تمامی شبکه‌ها برابر با ۱ است. این موضوع مطابق رابطه‌ی ۵، نشان می‌دهد که مقدار خطای بازسازی (MSE) تمامی نمونه‌های معیوب بیشتر از مقدار آستانه بوده و تعداد پیش‌بینی‌های منفی کاذب در همه‌ی شبکه‌ها مساوی صفر شده است (این موضوع در شکل ۱۱ نیز نشان داده شده است).

پس از طراحی معماری خودرمزگذار لازم است تمامی فرایندهای عیب‌یابی روی لبه پیاده‌سازی و یک دستگاه مستقل برای شناسایی ناهنجاری ساخته شود. در رابطه با آموزش خودرمزگذار روی لبه، چالش‌های اصلی، شامل: محدودیت حافظه و توان پردازشی هستند. تمام میکروکنترلرهای موجود در بازار دارای

جدول ۴. ماتریس درهم‌ریختگی.

Table 4. Confusion Matrix.

	Predicted Healthy	Predicted Faulty
Actual Healthy	599 (True Negative)	1 (False Positive)
Actual Faulty	0 (False Negative)	2800 (True Positive)

جدول ۵. عملکرد مدل براساس معیارهای ارزیابی.

Table 5. Model performance score.

Criteria	Precision	Recall	Accuracy	F1 Score
Score	0.9996	1	0.9997	0.9998

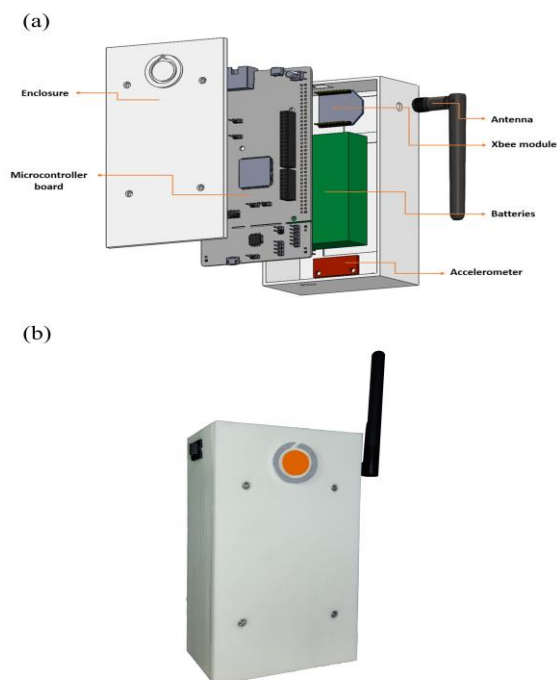


Figure 12. (a) Various components of the device, (b) Photo of the device.

شکل ۱۲. (a) اجزاء مختلف افزار توسعه یافته، (b) تصویری از دستگاه ساخته شده.

دسته‌ای ۴۲ قادر است رفتار ناهنجار ماشین را به درستی بازیابی و دقت و امتیاز F1 بالای ۹۹/۹٪ را شناسایی کند.

۴. معماری دستگاه

با برآوردی از میزان حافظه‌ی لازم برای اجرای کل الگوریتم و توان پردازشی موردنیاز برای آموزش شبکه، میکروکنترلر STM32H743ZI2 به‌عنوان یک گزینه‌ی ارزان‌قیمت و با عملکرد بالا برای پیاده‌سازی الگوریتم انتخاب شده است تا تجهیز بر پایه‌ی آن توسعه داده شود. دستگاه توسعه داده شده شامل یک برد میکروکنترلر STM32H743ZI2 با هسته‌ی Arm Cortex-M7، یک شتاب‌سنج دیجیتال سه‌محوره‌ی LIS3DSH، دو باتری لیتیوم-یونی، یک مازول بی‌سیم اکس‌بی، آنتن مازول اکس‌بی، و یک محفظه است (شکل ۱۲-).

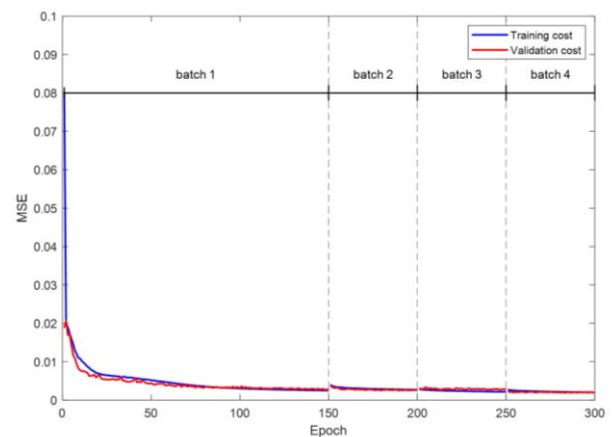


Figure 10. Training and validation costs in batch-based training.

شکل ۱۰. مقادیر تابع هزینه بر روی مجموعه‌ی داده‌های آموزش و راستی‌آزمایی.

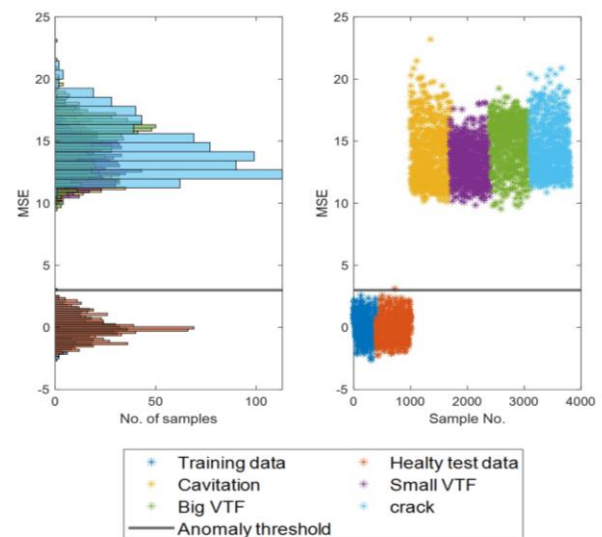


Figure 11. Distribution of normal and abnormal data.

شکل ۱۱. توزیع خطای میانگین مربعات داده‌های سالم و ناهنجار.

همچنین در شکل ۱۱، خطای میانگین مربعات داده‌های عادی و ناهنجار مشاهده می‌شود؛ که مطابق توضیحات اخیر، مقادیر MSE داده‌های آموزشی نرمال‌سازی و سطح سه-سیگما به‌عنوان آستانه‌ی ناهنجاری انتخاب شده است. مشاهده می‌شود که مقادیر MSE نمونه‌های معیوب به‌طور قابل توجهی بالاتر از خط آستانه قرار گرفته‌اند. این امر نشان می‌دهد که شبکه به خرابی‌ها، حساسیت بالایی دارد.

همچنین ماتریس درهم‌ریختگی مربوط به آزمون نمونه‌های سالم و معیوب در جدول ۴ ارائه شده است. فقط یک نمونه‌ی داده‌ی سالم کمی بالاتر از خط آستانه قرار گرفته است. با این حال هنگامی که عیب در پمپ ایجاد شده است، مدل توانسته است با موفقیت، مقدار MSE بسیار بالاتری را به آن اختصاص دهد.

همان‌طور که در جدول ۵ نشان داده شده است، خودرمزگذار با معماری ارائه شده در جدول ۳، نرخ یادگیری ۰/۰۰۱ و با استفاده از روش یادگیری

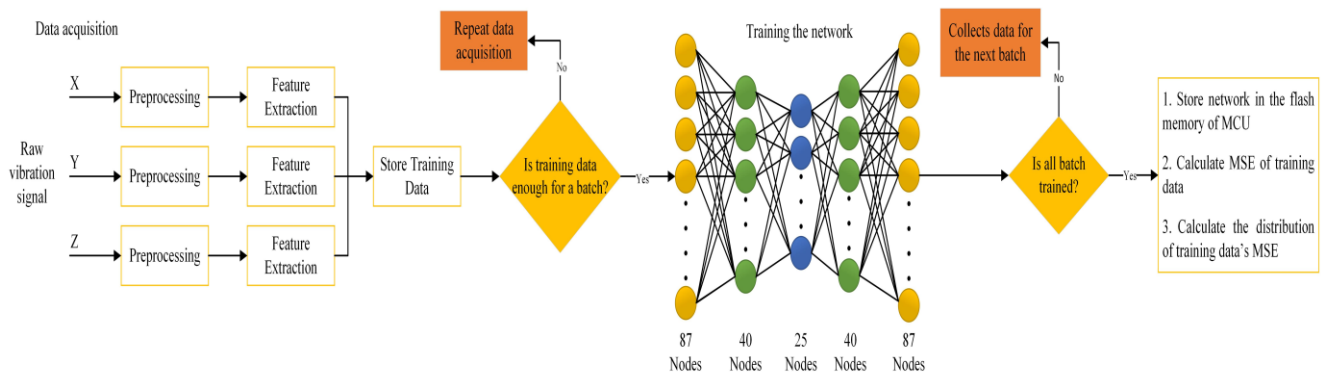


Figure 13. The algorithm flowchart for edge computing.

شکل ۱۳. فلوچارت الگوریتم رایانش لبه‌ای.

میکروکنترلر برای استفاده‌های آتی ذخیره می‌شود. خطای میانگین مربعات تمام دسته‌ها برای محاسبه‌ی توزیع داده‌ها در شرایط سالم ماشین جهت نرمال‌سازی نمونه‌های بعدی به‌کار می‌رود. پس از تکمیل مراحل آموزش، ۸۷ ویژگی از یک نمونه‌ی جدید وارد مدل می‌شود و میکروکنترلر خطای میانگین مربعات بین نمونه و بازتولیدشده‌ی آن را محاسبه می‌کند تا مشخص شود آیا مقدار مذکور از آستانه‌ی ناهنجاری بیشتر است یا خیر. اگر وضعیت ماشین سالم تشخیص داده شود، یعنی MSE کمتر از حد آستانه‌ی ناهنجاری باشد، میکروکنترلر و شتاب‌سنج به حالت خواب می‌روند و پس از چند دقیقه، بیدار می‌شوند و وضعیت ماشین را مجدداً ارزیابی می‌کنند.

در صورتی که MSE نمونه‌ی جدید از آستانه‌ی ناهنجاری بیشتر شود، ماژول اکسی‌بی بیدار می‌شود و یک هشدار به همراه داده‌های ارتعاشی خام را از طریق پروتکل زیگ‌بی^{۴۳} به اتاق نظارت ارسال می‌کند تا کارشناسان وضعیت ماشین را بررسی کنند. این فرآیند به‌طور خودکار توسط میکروکنترلر هر چند دقیقه یک‌بار در طول عمر ماشین انجام می‌شود. به‌منظور اثبات تحقق‌پذیری فناوری اخیر، آزمایش دیگری بر روی سیستم انجام شده است، که طی آن تمامی مراحل شرح داده‌شده، از جمله جمع‌آوری دسته‌های داده، آموزش شبکه، و آزمون با داده‌های جدید به‌طور خودکار توسط میکروکنترلر اجرا شده است. پس از آموزش مدل با ۴ دسته‌ی داده، شامل ۱۰۰ نمونه‌ی داده‌ی سالم در ۱۰ نقطه‌ی عملکردی مختلف پمپ، ۸۶ نمونه‌ی سالم دیده‌نشده به شبکه ارائه شده است. سپس کاپیتاسیون شبیه‌سازی و این بار ۵۷ آزمون در شرایط سالم پمپ انجام شده است. تعداد آزمون‌ها تحت شرایط وجود کاپیتاسیون نیز ۱۰۰ بوده است. مقادیر MSE تمام داده‌ها و هزینه‌های آموزش و راستی‌آزمایی هر دسته در حافظه‌ی فلش ذخیره شد تا برای ترسیم شکل ۱۴ استفاده شوند.

همانند شکل ۱۰، در شکل ۱۴(a) نیز هزینه‌های آموزش و راستی‌آزمایی با آموزش خودرمزگذار با دسته‌های بیشتر کاهش یافته است، که نشان‌دهنده‌ی این واقعیت است که مدل با داده‌های سالم دچار بیش‌برازش نشده است. معیارهای درستی، بازیابی، دقت، و امتیاز F1 همگی مساوی ۱ هستند.

علاوه‌بر این، مطابق ۱۴(b)، شبکه با موفقیت بر روی میکروکنترلر آموزش دیده و توانسته است شرایط سالم و ناهنجار ماشین را تشخیص دهد. همان‌طور که مشخص است، اختلاف قابل‌توجهی میان مقادیر MSE نمونه‌های سالم و معیوب وجود دارد. در حالی که مقادیر MSE سیگنال‌های سالم همگی کمتر

(a). میکروکنترلر با بسامد ۴۸۰ مگاهرتز کار می‌کند و ۱ مگابایت رم و ۲ مگابایت حافظه‌ی فلش برای ذخیره‌ی کد دارد. شتاب‌سنج دیجیتال، سیگنال‌های ارتعاشی را در سه محور با بسامد نمونه‌برداری ۱۶۰۰ هرتز و رزولوشن ۱۶ بیت دریافت می‌کند و به یک فیلتر آنتی-آلیسینگ با بسامد گوشه‌ی ۸۰۰ هرتز مجهز است. ماژول اکسی‌بی می‌تواند داده‌ها را با نرخ ۲۵۰ هزار بیت بر ثانیه در یک دایره به شعاع تقریبی ۹۰ متر ارسال و دریافت کند.

برای تخمین مدت زمان عملکرد مستقل دستگاه با باتری، مصرف انرژی میکروکنترلر در دو حالت کاری مختلف در نظر گرفته شده است. طبق مستندهای فنی ارائه‌شده برای خانواده‌ی STM32H7، جریان مصرفی این میکروکنترلر در حالت فعال (Run Mode) حدود ۱۱۰ میلی‌آمپر (در بسامد ۴۸۰ مگاهرتز) و در حالت کم‌مصرف (Stop Mode) حدود ۰/۱۵ میلی‌آمپر است.^[۲۴] میکروکنترلر به‌صورت پیش‌فرض در حالت کم‌مصرف قرار دارد و فقط هنگام نمونه‌برداری و پردازش داده‌ها به‌مدت تقریباً ۵ ثانیه از حالت اخیر خارج می‌شود. با فرض انجام نمونه‌برداری هر ۳۰ دقیقه یک‌بار، در مجموع ۴۸ بار در شبانه‌روز میکروکنترلر فعال خواهد بود.

بر این اساس، با استفاده از دو باتری لیتیوم‌یون هر کدام با ظرفیت ۴۰۰۰ میلی‌آمپر-ساعت، عمر تقریبی سامانه در شرایط عملیاتی حدود ۲ سال برآورد می‌شود. این تخمین مبتنی بر ترکیب زمان‌بندی چرخه‌های کاری و مقادیر جریان مصرفی در هر حالت بوده و به‌عنوان برآوردی اولیه از قابلیت عملکرد بلندمدت دستگاه بدون نیاز به تعویض یا شارژ مجدد باتری ارائه شده است. تصویری از دستگاه ساخته‌شده‌ی نهایی در شکل ۱۲(b) ارائه شده است.

۵. یادگیری برخط مبتنی بر رایانش لبه‌ای

در بخش کنونی، جزئیات یادگیری برخط الگوریتم و چارچوب تشخیص خودکار ناهنجاری ارائه شده است. در شکل ۱۳، خط سیر داده‌ها برای رایانش لبه‌ای مشاهده می‌شود. شتاب‌سنج، داده‌های ارتعاشی خام را از دستگاه دریافت و در آرایه‌ای با ابعاد ۱×۲۰۴۸ ذخیره می‌کند. پس از حذف مؤلفه‌ی دی‌سی، ۲۹ ویژگی از هر محور استخراج می‌شود. آرایه‌های ویژگی مذکور به یکدیگر الحاق می‌شوند و آرایه‌ای با ابعاد ۱×۸۷ را تشکیل می‌دهند، که یک نمونه‌ی داده را می‌سازد. با ذخیره‌ی ۱۰۰ نمونه در یک ماتریس، اولین دسته برای آموزش خودرمزگذار آماده می‌شود. این فرآیند تا زمان استفاده از هر چهار دسته در آموزش شبکه ادامه می‌یابد. در پایان، مدل آموزش‌دیده در حافظه‌ی فلش

۶. نتیجه‌گیری

روش‌های یادگیری ماشین و هوش مصنوعی برای حل مسائل صنعتی در سال‌های اخیر رشد چشمگیری داشته‌اند. آموزش الگوریتم‌های ذکرشده بر روی میکروکنترلر برای دستیابی به راه‌حل‌های کارآمد، امری ضروری است، به‌ویژه در کاربردهای پایش وضعیت که مدل‌های آموزش‌دیده روی لبه از شبکه‌های از پیش آموزش‌دیده بهتر عمل می‌کنند. [۱۸] با این حال، آموزش روی دستگاه حتی امروزه مسئله‌ای چالش‌برانگیز است. [۱۷-۱۳] در نوشتار حاضر، یک روش نوآورانه‌ی پردازش در لبه و یادگیری دسته‌ای برای رفع مشکل اخیر پیشنهاد شده است. انگیزه‌ی توسعه‌ی چنین چارچوب یادگیری بدون نظارت و برخط برای تشخیص ناهنجاری در ماشین‌های دوار بدین شرح است:

(۱) در روش‌های با نظارت به حجم وسیعی از داده‌های مربوط به وضعیت سالم و معیوب ماشین نیاز است. به‌وضوح جمع‌آوری داده‌های معیوب برای هر ماشینی که نیاز به پایش دارد، در صنعت غیرعملی است.

(۲) آموزش آفلاین روش‌های بدون نظارت روی کامپیوترهای شخصی برای عیب‌یابی کارآمد نیست. هر ماشین مشخصه‌های منحصره‌فردی دارد و مدلی که با داده‌های یک ماشین خاص آموزش دیده است، فقط برای همان ماشین کاربرد دارد. بنابراین، به جمع‌آوری داده‌های منحصر به هر ماشین، آموزش یک مدل خاص برای هر یک و استفاده از هر مدل برای عیب‌یابی منحصر به آن ماشین پرداخته شده است. این فرایند، به‌ویژه زمانی که نیاز به پایش تعداد زیادی تجهیز باشد، بسیار دشوار و زمان‌بر است.

(۳) رایانش ابری بلادرنگ می‌تواند راه‌حلی برای این مسئله باشد، اما این رویکرد بسیار پرهزینه است و انتقال حجم زیادی از داده‌ها از نزدیکی ماشین‌ها به سرورهای ابری چالشی بزرگ محسوب می‌شود. این انتقال در مقیاس‌های بزرگ و صنعتی، نیازمند سرورهایی با پهنای باند و سرعت انتقال داده‌ی بالا و قدرت پردازشی قابل‌توجه است، که خود مانعی اساسی در اجرای این رویکرد به شمار می‌آید.

در پژوهش حاضر به‌منظور غلبه بر این محدودیت‌ها و چالش‌ها، از تعدادی ویژگی شناخته‌شده برای شناسایی ناهنجاری در ماشین‌آلات دوار استفاده شده و یک خودرمزگذار سبک با سه لایه‌ی مخفی و الگوریتم پس‌انتشار خطا برای آموزش آن به زبان برنامه‌نویسی C توسعه یافته است. برای جمع‌آوری داده، یک سیستم متشکل از پمپ گریز از مرکز با سیستم داده‌برداری مناسب طراحی شده است. پس از آزمایش موفقیت‌آمیز مدل به‌صورت آفلاین روی این سیستم صنعتی با استفاده از رایانه، کل مدل به همراه الگوریتم آموزشی آن به یک میکروکنترلر منتقل و افزاری مستقل برای عیب‌یابی ماشین‌های دوار توسعه داده شده است. این دستگاه شامل برد توسعه‌ی میکروکنترلر ارزان‌قیمت با عملکرد بالا از شرکت ST، شتاب‌سنج دقیق، باتری‌های لیتیوم‌یونی که امکان کارکرد آن را به مدت حدود ۲ سال بدون نیاز به شارژ مجدد فراهم می‌کنند، و ماژول بی‌سیم اکس‌بی برای ارسال وضعیت بلادرنگ ماشین به اپراتور است. نتایج نشان داده است که دستگاه مذکور قادر به آموزش شبکه مانند رایانه است. حتی نتایج مدل آموزش‌دیده بر روی میکروکنترلر، بهتر از نتایج آن روی رایانه بوده است. البته فرایند آموزش روی میکروکنترلر زمان بیشتری می‌برد، که با توجه به تفاوت قدرت پردازش رایانه و میکروکنترلر، امری قابل‌درک است.

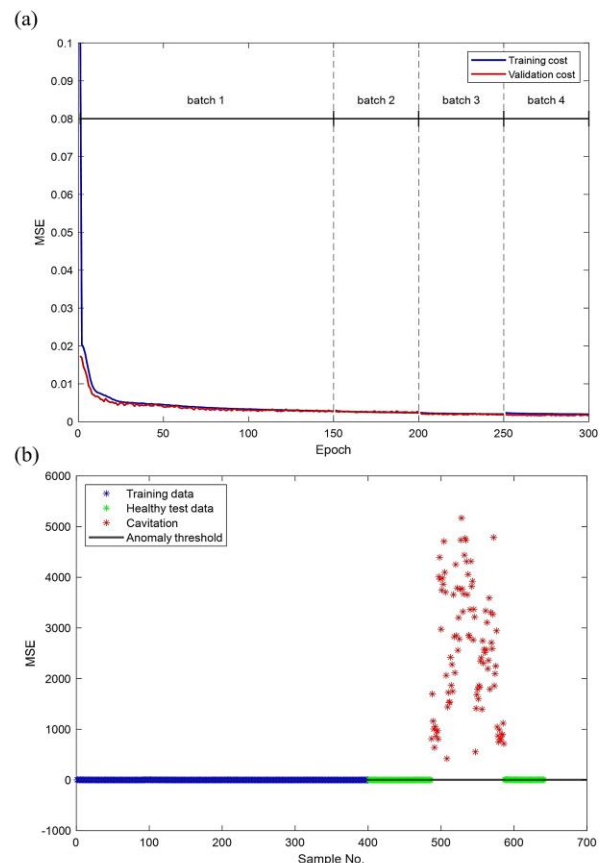


Figure 14. (a) Training and validation losses of the network while training on MCU and (b) The healthy and faulty data MSEs after training the autoencoder on the MCU.

شکل ۱۴. (a) هزینه‌ی آموزش و راستی‌آزمایی در طول آموزش با میکروکنترلر، (b) MSE داده‌های سالم و ناهنجار در شبکه‌ی آموزش دیده با میکروکنترلر.

جدول ۶. زمان کلی آموزش شبکه.

Table 6. The total training time.

	PC	MCU
Time (s)	3.095	466.02

از مقدار ۳ هستند، کمینه‌ی مقدار MSE داده‌های معیوب حدود ۵۰۰ است. این امر نشان می‌دهد که شبکه‌ای که بر روی میکروکنترلر آموزش دیده است، حتی حساس‌تر از شبکه‌ای است که روی کامپیوتر آموزش دیده است.

بنابراین، می‌توان نتیجه گرفت که الگوریتم نه فقط قادر به شناسایی عیوب معمول در این پمپ گریز از مرکز است، بلکه می‌تواند ناهنجاری‌های بسیار کوچک و جزئی در ماشین‌های دوار را نیز تشخیص دهد. این ناهنجاری‌های جزئی می‌توانند شامل ترک‌های کوچک در پروانه، عیوب ناچیز در یاتاقان‌ها، یا هرگونه عیب جزئی دیگری باشند، که شرایط عادی عملکرد ماشین را تغییر می‌دهند. در جدول ۶، مقایسه‌ی زمان کل آموزش چهار دسته‌ی داده در کامپیوتر و میکروکنترلر ارائه شده است. مشخص است که زمان آموزش شبکه‌ی مشابه در میکروکنترلر تفاوت چشمگیری با زمان آموزش آن در رایانه‌ی شخصی دارد، زیرا توان پردازشی یک میکروکنترلر کوچک بسیار ضعیف‌تر است.

منابع - References

- Ding, S.X., 2008. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media. <https://doi.org/10.1007/978-3-540-76304-8>
- Khan, S. and Yairi, T., 2018. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107, pp.241-265. <https://doi.org/10.1016/j.ymssp.2017.11.024>
- Wu, D., Liu, S., Zhang, L., Terpenney, J., Gao, R.X., Kurfess, T. and Guzzo, J.A., 2017. A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing. *Journal of Manufacturing Systems*, 43, pp.25-34. <https://doi.org/10.1016/j.jmsy.2017.02.011>
- Xenakis, A., Karageorgos, A., Lallas, E., Chis, A.E. and González-Vélez, H., 2019. Towards distributed IoT/cloud based fault detection and maintenance in industrial automation. *Procedia Computer Science*, 151, pp.683-690. <https://doi.org/10.1016/j.procs.2019.04.091>
- Park, D., Kim, S., An, Y. and Jung, J.Y., 2018. LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks. *Sensors*, 18(7), p.2110. <https://doi.org/10.3390/s18072110>
- Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L., 2016. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5), pp.637-646. <https://doi.org/10.1109/JIOT.2016.2579198>
- Qian, G., Lu, S., Pan, D., Tang, H., Liu, Y. and Wang, Q., 2019. Edge computing: A promising framework for real-time fault diagnosis and dynamic control of rotating machines using multi-sensor data. *IEEE Sensors Journal*, 19(11), pp.4211-4220. <https://doi.org/10.1109/JSEN.2019.2899396>
- Huang, H., Yang, L., Wang, Y., Xu, X. and Lu, Y., 2021. Digital twin-driven online anomaly detection for an automation system based on edge intelligence. *Journal of Manufacturing Systems*, 59, pp.138-150. <https://doi.org/10.1016/j.jmsy.2021.02.010>
- Yang, H., Mathew, J. and Ma, L., 2002. Intelligent diagnosis of rotating machinery faults-a review. In *Proceedings of the 3rd Asia-Pacific Conference on Systems Integrity and Maintenance*, pp. 385-392. Queensland University of Technology Press.
- Markou, M. and Singh, S., 2003. Novelty detection: a review—part 2:: neural network based approaches. *Signal processing*, 83(12), pp.2499-2521. <https://doi.org/10.1016/j.sigpro.2003.07.019>
- Maasoum, S.M.H., Mostafavi, A. and Sadighi, A., 2020, December. An autoencoder-based algorithm for fault detection of rotating machines, suitable for online learning and standalone applications. In *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1-6. IEEE. <https://doi.org/10.1109/ICSPIS51611.2020.9349574>
- Ahmad, S., Styp-Rekowski, K., Nedelkoski, S. and Kao, O., 2020, December. Autoencoder-based condition monitoring and anomaly detection method for rotating machines. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 4093-4102. IEEE. <https://doi.org/10.1109/BigData50022.2020.9378015>
- A simple, fast and optimized method to deploy Neural Networks on STM32: https://www.st.com/content/st_com/en/about/innovation---technology/artificial-intelligence.html
- TensorFlow Lite for Microcontrollers: <https://www.tensorflow.org/lite/microcontrollers>
- CMSIS NN software library: <https://www.keil.com/pack/doc/CMSIS/NN/html/index.html>
- X-CUBE-AI, AI expansion pack for STM32CubeMX: <https://www.st.com/en/embedded-software/x-cube-ai.html>
- Apache TVM: <https://tvm.apache.org/>
- Ren, H., Anicic, D. and Runkler, T.A., 2021, July. Tinyol: Tinyml with online-learning on microcontrollers. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. IEEE. <https://doi.org/10.1109/IJCNN52387.2021.9533927>
- Tandon, N. and Choudhury, A., 1999. A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. *Tribology International*, 32(8), pp.469-480. [https://doi.org/10.1016/S0301-679X\(99\)00077-8](https://doi.org/10.1016/S0301-679X(99)00077-8)
- Williams, T., Ribadeneira, X., Billington, S. and Kurfess, T., 2001. Rolling element bearing diagnostics in run-to-failure lifetime testing. *Mechanical Systems and Signal Processing*, 15(5), pp.979-993. <https://doi.org/10.1006/mssp.2001.1418>
- Xia, Z., Xia, S., Wan, L. and Cai, S., 2012. Spectral regression based fault feature extraction for bearing accelerometer sensor signals. *Sensors*, 12(10), pp.13694-13719. <https://doi.org/10.3390/s121013694>
- Goumas, S.K., Zervakis, M.E. and Stavrakakis, G.S., 2002. Classification of washing machines vibration

- signals using discrete wavelet analysis for feature extraction. *IEEE Transactions on Instrumentation and Measurement*, 51(3), pp.497-508.
<https://doi.org/10.1109/TIM.2002.1017721>.
23. CodeBlocks C compiler,
<https://www.codeblocks.org/>.
24. STMicroelectronics, "STM32H742xI/G, STM32H743xI/G: 32-bit Arm® Cortex®-M7 480MHz MCUs, up to 2MB Flash, up to 1MB RAM, 46 com. and analog interfaces,"
<https://www.st.com/resource/en/datasheet/stm32h743zi.pdf>

این صفحه عمل خالی گذاشته شده است.