Optimizing LSTM-based temperature prediction algorithm for embedded system deployment

Original
Optimizing LSTM-based temperature prediction algorithm for embedded system deployment / D'Agostino, Pietro; Violante, Massimo; Macario, Gianpaolo. - ELETTRONICO. - (2024), pp. 01-07. (Intervento presentato al convegno 29TH INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION tenutosi a Padova (IT) nel September 10-13, 2024) [10.1109/ETFA61755.2024.10711142].

Availability:
This version is available at: 11583/2993014 since: 2024-10-22T12:48:34Z

Publisher:
IEEE

Published
DOI:10.1109/ETFA61755.2024.10711142

(Article begins on next page)

Optimizing LSTM-based temperature prediction algorithm for embedded system deployment

Pietro d'Agostino
DAUIN, Computer and control department
Politecnico di Torino
Turin, Italy
pietro.dagostino@polito.it

Massimo Violante
DAUIN, Computer and control department
Politecnico di Torino
Turin, Italy
massimo.violante@polito.it

Gianpaolo Macario
R&D
AROL Closure Systems
Canelli, Italy
gianpaolo.macario@arol.com

***Abstract*—This paper presents a flexible Industrial Internet of Things (IIoT) infrastructure model, highlighting the integration of the Long Short-Term Memory (LSTM) algorithm for predictive analysis. A key component of this model is the Concentrator, a fog-computing local hub that provides a sandbox environment for third-party developments. Within this framework, clients can collect data using mechanisms provided by Original Equipment Manufacturers (OEMs), such as AROL Closure Systems, while independently developing proprietary algorithms. This approach eliminates the need for direct interaction with OEMs. The paper explores the use of the LSTM algorithm to develop a predictor for analyzing machine temperature behavior, allowing for the anticipation of potential faults.**

***Index Terms*—Predictive maintenance, Industry 4.0, Fog computing, Deep learning, LSTM, Embedded systems, Raspberry Pi, Temperature prediction**

I. Introduction

In the rapidly evolving landscape of Industry 4.0, integrating cutting-edge technologies has become important for enhancing operational efficiency and reliability. One of the main aspects of this industrial revolution is predictive maintenance, a paradigm shift from traditional reactive approaches towards a proactive and data-driven strategy [1]. This methodology leverages real-time data to forecast equipment failures and optimize maintenance schedules, minimizing downtime and reducing operational costs [2]. Industry 4.0 has ushered in a new era of intelligent manufacturing, characterized by the seamless integration of cyber-physical systems, the Internet of Things (IoT), and advanced data analytics [3] [4].

Integrating machine learning techniques significantly enhances maintenance systems by extracting insights from vast sensor data [5]. Supervised learning algorithms like Support Vector Machines (SVM) and Random Forests excel in deciphering intricate patterns [6]. Trained on historical data, these algorithms identify normal behavior and deviations signaling potential faults, empowering proactive maintenance actions [7] [8]. In tandem with supervised learning, unsupervised techniques are useful in predictive maintenance. Clustering and anomaly detection unveil hidden patterns and anomalies in data streams sans labeled training data [9]. Furthermore, advanced analytics platforms and cloud or fog computing have revolutionized the scalability and accessibility of predictive maintenance solutions [10]. These platforms provide the computational muscle and storage capacity to crunch large real-time sensor data volumes, democratizing predictive modeling and allowing organizations of all sizes to seamlessly deploy solutions across their infrastructure [11]. By harnessing the potential of big data analytics and artificial intelligence, predictive maintenance promises to shift the maintenance paradigm from reactive to proactive, allowing unprecedented efficiency, reliability, and performance levels. Organizations can optimize maintenance schedules, prioritize critical assets, and allocate resources more effectively using predictive insights. Ultimately, the transition to predictive maintenance signifies a fundamental shift from industrial systems management to an era of intelligent, efficient asset management practices [12].

In this ever-evolving landscape of data-driven decision-making and IoT paradigms, time series forecasting emerges as a cornerstone for unraveling complex temporal patterns and predicting future trends. [13]. Time series data, characterized by its temporal nature, encapsulates a wealth of information essential for informed decision-making. Accurate forecasting empowers organizations to proactively respond to trends, mitigate risks, and optimize strategies [14]. The rapid evolution of computational capabilities and sophisticated machine-learning techniques have propelled time series forecasting into a new era. Recent advancements encompass a spectrum of methodologies, including deep learning architectures, ensemble techniques, and hybrid models that synergize classical statistical approaches with cutting-edge algorithms [15].

This paper presents a real-based application scenario employing the Concentrator, a component of a fog computing system serving as a local hub within a production line for gathering data from various wireless devices throughout the structure. Our study centers on a production line featuring machines dedicated to bottle cleaning, filling, capping, and labeling. Notably, we highlight the capping machines, the primary product of AROL Closure Systems, the company affiliated with the authors. These intricate mechanical systems

AROL Closure Systems

are prone to faults due to prolonged usage and mechanical complexities, necessitating proactive maintenance measures. The Concentrator operates within the system's personal area network (PAN), gathering data to generate predictive maintenance reports and discerning potential machine faults based on the collected information. Leveraging a sandbox environment provides a platform for third-party algorithm development, enabling the creation of proprietary algorithms independently without direct interaction with the Original Equipment Manufacturers (OEM).

II. State of Art

We aim to seamlessly integrate a time series forecasting algorithm into an embedded product, positioning it as a local hub within a fog computing system. The primary objective of this algorithm is to predict sensor values on a machine, thereby facilitating proactive measures for predictive maintenance, optimizing system performance, and understanding if it can be applied in a real scenario with a real working machine.

In the study presented in [16], the authors comprehensively compare various models for anomaly detection in computer networks. Leveraging time series forecasting methods, they meticulously evaluate the effectiveness of Auto-Regressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), FB Prophet, and Long Short-Term Memory (LSTM) models. Through experimentation, they identify LSTM as the superior model and underscore its performance in minimizing error rates, especially in the context of computer network anomaly detection.

Moving on to [17], the focus shifts towards refining time series predictions by addressing anomalies. The researchers propose a meticulous approach involving dataset differencing, anomaly identification, and subsequent thresholding through the computation of mean values. Notably, anomalies are effectively eliminated by adjusting preceding values based on the magnitude of the identified anomaly. This method enhances the precision and reliability of time series predictions, especially when dealing with dynamic and fluctuating datasets.

In a different application domain, the research outlined in [18] harnesses a Deep Long Short-Term Memory (LSTM) encoder-decoder architecture and an IoT sensor structure in coal mines. The primary goal is to predict critical parameters such as wind speed, CH4 concentration, and potential fire hazards. Utilizing Zigbee modules, the system successfully indicates accidents with an impressive accuracy rate of 94.23 %, providing a valuable tool for enhancing safety in critical environments.

[19] introduces an innovative approach to tracking food waste, opting for machine learning models over traditional IoT methods. The authors employ the ARIMA model to predict food wastage and fog computing technology to enhance response time, security, and overall system efficiency. This alternative approach showcases the versatility of machine learning models in diverse applications, outperforming traditional IoT prediction rates and efficiency.

Adopting machine learning algorithms and time series analysis for predictive maintenance in IoT systems was reviewed in [20]. This comprehensive study charted the progression from the initial to the fourth industrial revolution, highlighting emerging technologies such as IoT, AI, cloud computing, and big data analytics in converting time series data into actionable insights.

In real-time IoT sensor data, [21] presented a system for identifying potential failures through advanced machine learning to prevent disruptions and enhance production efficiency. Noteworthy for its integration into real-world manufacturing, the paper evaluated high-performing machine learning models, including ensemble methods like Random Forest and boosting techniques like XGBoost.

Finally, the intricate landscape of predictive maintenance in highly automated production lines was investigated in [22], explicitly focusing on automobile part manufacturing machines. Diverging from traditional maintenance approaches, predictive maintenance proactively anticipates equipment failures to minimize downtime. The study proposed an intelligent approach featuring a weight-optimized Gated Recurrent Unit (GRU) model and the Whale Optimization with Seagull Algorithm for improved accuracy in Predictive Maintenance planning, offering efficient solutions for predicting future component failures in mechanical part-making machines.

After carefully considering various works and studies, we ultimately implemented the Long Short-Term Memory (LSTM) algorithm. Our decision stemmed from its perceived stability and efficiency compared to alternative solutions. While it is widely utilized, one notable drawback is its demand for computational resources. However, this challenge can be mitigated by selecting appropriate embedded hardware and fine-tuning the model weights.

III. Methodology and Architecture

Building upon our prior research outlined in papers [23] and [24], we propose an advanced general system, shown in Fig. 1, characterized by its distinct hardware foundation and the integration of a time series forecasting algorithm for predictive analytics. Within the industrial fog computing system, the Concentrator acts as a local central hub, leveraging virtualization to establish a sandbox for external users to develop custom applications. We aim to enhance the default system by introducing the capability to predict future outcomes within an embedded system constrained by limited resources.

The proposed general system comprises three primary components, adhering to the structure delineated in [25]:

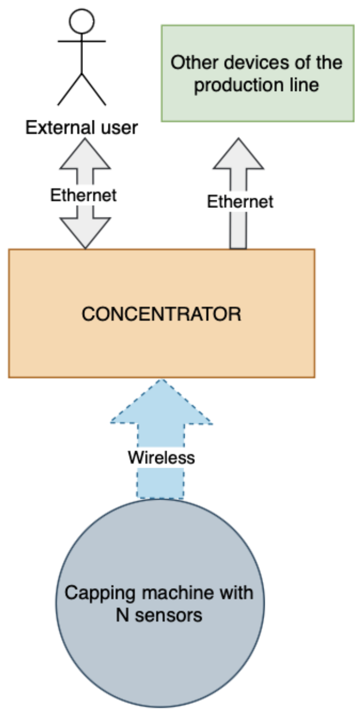

Fig. 1. General System

A. The Capping Machine (PAN)

This component encapsulates the Personal Area Network (PAN) within the fog computing system, serving as the edge environment where sensors are strategically positioned to collect and analyze crucial data. The capping machine, equipped with variable heads depending on size, incorporates sensors on each unit. These sensors study the behaviors of each unit, providing a granular understanding of the machine's performance within the production line. This sensor network can be enhanced using AROL's native and external sensors from the customer, offering a holistic perspective on the capping machine's operational dynamics.

B. The Concentrator (LAN)

The Concentrator emerges as the Local Area Network (LAN) within the fog computing system, functioning as a local central hub. Its multifaceted role involves aggregating data from various sensors, subsequent elaboration, and transmission to external users or other parts of the production line. This comprehensive approach optimizes data exchange, accelerates response times, and significantly bolsters system security and usability [26]. The Concentrator supports diverse communication methods, employing wireless communication within the PAN and Ethernet communication with the broader system. A key component of this local central is the virtualization through Docker, which, through its containers, lets the producer create a sandbox for the host to apply its code and produce its elaboration, also adding additional sensors into the system, alongside privacy and security ensure.

C. External Users (WAN)

External users form an integral part of the Wide Area Network (WAN), crucial in configuring the Concentrator, managing updates, and initiating various data elaborations. Their engagement is essential for tailoring the system to specific requirements and ensuring seamless functionality. On the other hand, the broader system, encompassing the remaining production line and associated devices, capitalizes on the generated reports for predictive maintenance. This strategic utilization of data empowers the system to proactively address potential issues and anomalies while notifications relay pertinent information regarding operational situations encountered by various devices.

IV. Experiment and Discussion

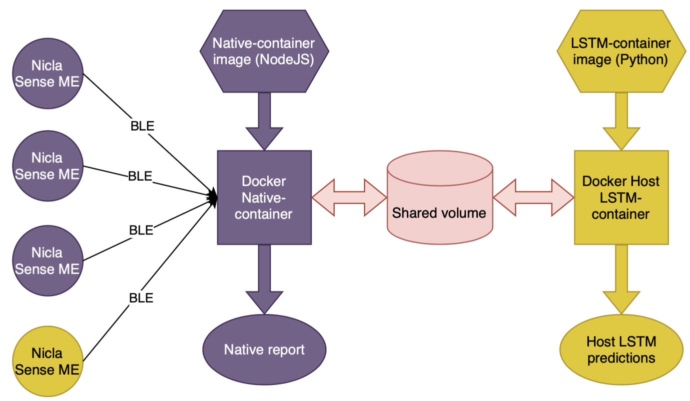

Fig. 2. The Docker containers

This section is dedicated to describing the experiment and the implemented scenario, shown in Fig. 2 to prove the efficiency and feasibility of the proposed system above. The system is composed of the following parts:

A. The peripherals: The Nicla Sense ME

The Nicla Sense ME embodies a trifecta of virtues: compactness, affordability, and energy efficiency, while seamlessly integrating four cutting-edge sensors developed by Bosch Sensortec. Named with the acronym ME, representing "Motion" and "Environment," this device accurately detects rotations, accelerations, temperatures, humidity, pressure, air quality, and CO2 levels with industrial-grade precision. This latest iteration sets a new standard as the smallest form factor yet, enabling effortless data transmission via Bluetooth Low Energy connectivity (version 4.2), empowered by an ANNA-B112 module. Only four Nicla devices were utilized in the experiment: three were designated as Native and one as the Host. Their primary function is to relay temperature, humidity, and pressure readings to the concentrator through the wireless protocol.

B. The Concentrator: The Raspberry Pi 4 Model B

We harnessed the capabilities of an embedded hardware system: the Raspberry Pi 4 Model B. Operating on Raspberry Pi OS 64-bit, built upon Debian 12 (bookworm), it served as our platform of choice. For virtualization and sandboxing purposes catering to external users, we leveraged Docker version 25.0.4, build 1a576c5. Docker Compose, in version 2.24.7, seamlessly facilitated concurrent multiple container management. Given our reliance on Bluetooth for wireless communication, we integrated BlueZ library version 5.66 into our setup. The system architecture is bifurcated into two entities: the Native and the Host. The Native segment underscores the manufacturer's baseline distribution, providing default system functionalities. Conversely, the Host segment acts as a sandbox environment for generic users. Here, not only can users upload their code, but they can also integrate additional sensors into the system to expand its measurement capabilities.

C. The Docker Native Container

The Concentrator's primary objective is to gather data from many peripherals. Employing a Node. js-based container powered by Node.js version 16 and leveraging the open-source library node-ble, this system intelligently discerns between Host and Native configurations solely based on peripheral addresses and configurations. It establishes seamless connections with the designated peripherals and meticulously records the data stream from the Nicla Sense ME sensors. The acquisition frequency is tuned to capture measurements at a rate of one per second. Upon successfully establishing connections and acquiring the requisite data, the container publishes the raw data locally using HTTP and Express libraries. The Native container utilizes the node-ble library and Bluetooth connection to publish native measurements. Every file line comprises the Epoch it was taken, its ID, the peripheral's MAC address, and the characteristic measured effective value.

D. The Docker Host LSTM Container

This container employs the LSTM machine learning model to forecast sensor temperature measurements. Built on 1019663 real data measurements and with 70% used for training and 30% used for testing, it used Python 3.11. It relies on several key libraries, including Numpy 1.26.3, Pandas 2.1.4, Ujson 5.4.0, TensorFlow 2.15.0, and sci-kit-learn 1.2.2. The use of TinyML was considered; however, a more conventional approach was selected since the chosen Raspberry Pi has sufficient memory and CPU capacity to handle TensorFlow. Its operation begins by parsing the raw file generated by the initial container, verifying sufficient data for prediction. Once the file meets the data threshold, it removes any NaN or zero values and identifies outliers within the 20 to 43 degrees Celsius temperature range.

Subsequently, it loads the pre-trained model dubbed "model_j0700.keras" and assesses the next sequential temperature measurements. The output includes the last recorded temperature in the file and the subsequent temperature prediction (e.g., temperature number 40 and the forthcoming temperature number 41). Predictions are executed using a window of 20 values, advancing with each new temperature reading, ultimately providing a singular forecasted value. The utilized model was trained using authentic data from the AROL Closure System's J0700 machine. Comprising three layers, it features one LSTM layer with 50 nodes and two Dense layers, one with 25 nodes and the other with a single node. It demonstrates an RMSE of 0.1838, with a prediction time of approximately 200 ms.

E. Execution of the experiment and analysis of the data

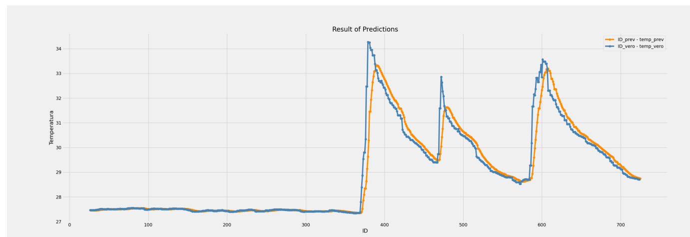

Fig. 3. First experiment: spikes

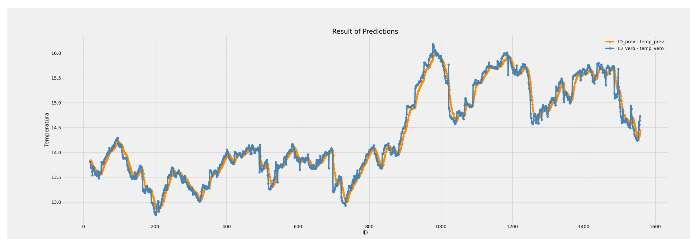

Fig. 4. Outdoor temperature mimicking experiment

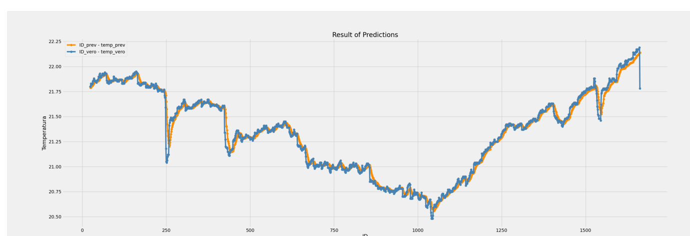

Fig. 5. Indoor temperature mimicking experiment

The experiments were conducted within a rigorously controlled environment, specifically a climatic room within the AROL facilities. The climatic room provided precise environmental conditions necessary for the experimentation, ensuring consistency and reliability throughout the testing process. Controlled manipulation involved deliberately heating the room to assess the responsiveness of the LSTM model. As illustrated in Fig. 3, the model demonstrated proficiency in forecasting fluctuating measurements, albeit encountering challenges with abrupt spikes in values.

This discrepancy can be attributed to the intrinsic characteristics of the trained model, model_j0700.keras, which underwent training on genuine machine data exhibiting a consistent environment devoid of such abrupt fluctuations. The Root Mean Square Error (RMSE) between the two curves was calculated to be 0.4502.

Subsequent evaluation involved utilizing authentic data to scrutinize the model's performance under real-world conditions. In this phase, the temperatures were set to mirror the outdoor ones, as depicted in Fig. 4.

The root mean square error (RMSE) for the two datasets is reported as 0.1155, and in a separate experiment, in which the temperature mirrored the indoor ones, as depicted in Fig. 5, it is reduced to 0.0469. These results underscore the model's commendable predictive precision across varying conditions. However, it's notable that the gradual temperature fluctuations observed in these scenarios do not fully represent the dynamics of sudden spikes or rapid fluctuations in temperature.

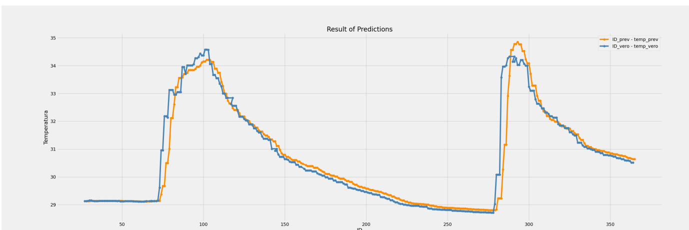

Fig. 6. Second experiment: spikes

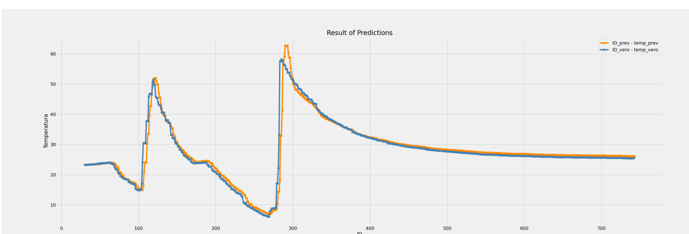

Fig. 7. Critical conditions

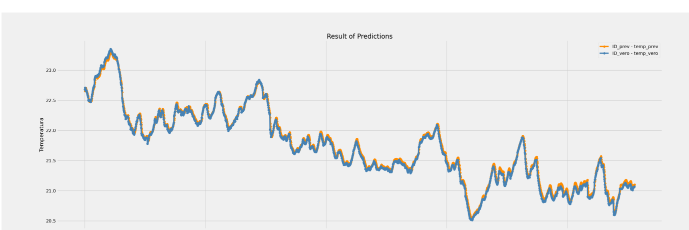

Fig. 8. Gradual temperature change

The study assessed the model's performance during sudden spikes, finding a notable disparity with RMSE exceeding 2.80. Spike patterns were integrated into the training dataset to enhance adaptability to volatile scenarios, maintaining the original model's configuration. This refinement significantly improved performance, reducing prediction time to 343ms.

Following the model refinement, two additional experiments were conducted to validate its enhanced performance. In the first experiment, an attempt to replicate the initial conditions yielded results depicted in Fig. 6, with an RMSE of 0.3895.

A controlled experimental setup was devised to induce stress on the sensor through extreme temperature variations. The effects on its performance were systematically analyzed by subjecting the sensor to freezing temperatures followed by rapid heating. The results of this experiment, as depicted in Fig. 7, revealed an RMSE (Root Mean Square Error) of 0.6735, indicating the sensor's response to abrupt temperature changes.

Subsequently, an outdoor mirroring deployment assessed the sensor's performance under real-world conditions with gradual temperature changes. The data collected from this deployment, illustrated in Fig. 8, showcased the sensor's reliability, yielding an RMSE of 0.0419. These findings were further consolidated through a comparative analysis summarized in Table I, highlighting the superior performance of the model_J0700_spikes.keras, particularly under prolonged real-world operating conditions resembling those encountered by industrial machinery.

TABLE I
RECAP OF THE EXPERIMENTS

Experiment	Model	RMSE
Slow spikes	model_J0700.keras	0.4502
Outdoors	model_J0700.keras	0.1155
Indoors	model_J0700.keras	0.0469
Verify model	model_J0700_spikes.keras	0.3895
Critical exp	model_J0700_spikes.keras	0.6735
Final outdoors	model_J0700_spikes.keras	0.0419

The table in Table II provides an insightful depiction of the system's response to the LSTM code implementation, aligning with the anticipated impact based on the model's weight. It shows the CPU and memory usage (max memory space is 7.63GB) for every stand-alone container, alongside the Raspberry basic system. The compose line, instead, the resources used for the entire system working simultaneously. Despite the observable strain on memory and CPU resources, the system maintains uninterrupted operation, indicating robust performance under load. Specifically, "Memory ON" denotes the quantitative memory utilization during container activation, whereas "Memory OFF" signifies memory status upon container shutdown.

TABLE II
SYSTEM CPU AND MEMORY

Name	CPU	Build Time	Memory ON	Memory OFF
Native	20%	142.2 s	750MB	660MB
LSTM	105.4%	322.2 s	1.01GB	766MB
Compose	101.3%	464.4 s	1.05GB	608MB
Raspberry	2%	/	/	585MB

V. CONCLUSIONS

This study highlights the important role of advanced algorithms in Industry 4.0, leveraging the Internet of Things (IoT), machine learning, and embedded systems to facilitate predictive maintenance. By employing these technologies, industrial systems can gain deeper insights into their products, enabling proactive problem-solving and preemptive maintenance interventions before machine faults occur. Utilizing a Long Short-Term Memory (LSTM) Algorithm has proven efficient and reliable for this purpose despite the challenges posed by limited hardware resources. It was embedded within a system employing virtualization through Docker to integrate this algorithm seamlessly. This approach enables the establishment of not only a Native system, where producers can showcase their products but also a sandbox environment tailored for individual customers. This sandbox incorporates diverse data processing techniques and sensor configurations to meet customer needs. Various experiments were conducted, including adjustments to the model employed, which showcased the system's feasibility without imposing undue strain on the hardware. These experiments underscored the reliability of the measurements, affirming the system's effectiveness. Consequently, the developed product stands poised for integration into a production line, primed for real-world deployment and utilization by discerning customers. In future work, this system will be tested on a real-case machine with at least 20 nodes to evaluate the flexibility of the choices made for the LSTM models, including the previously mentioned TinyML. Additionally, future studies will explore different training strategies, such as transfer learning, where the model is pre-trained on a broader dataset and then fine-tuned for specific operational scenarios encountered in the target industrial environment.

VI. Acknowledgments

The authors extend their heartfelt gratitude to AROL Closure System for generously sharing their expertise and resources, without which the fruition of this project would not have been possible. Additionally, sincere thanks are extended to our colleagues who provided invaluable assistance during the development of this paper.

References

[1] H. Toumi, A. Meddaoui and M. Hain, "The influence of predictive maintenance in industry 4.0: A systematic literature review," 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 2022, pp. 1-13, doi: 10.1109/IRASET52964.2022.9737901.

[2] A. Bousdekis, D. Apostolou and G. Mentzas, "Predictive Maintenance in the 4th Industrial Revolution: Benefits, Business Opportunities, and Managerial Implications," in IEEE Engineering Management Review, vol. 48, no. 1, pp. 57-62, 1 Firstquarter, march 2020, doi: 10.1109/EMR.2019.2958037.

[3] A. Sharma and A. Aslekar, "IOT Based Predictive Maintenance in Industry 4.0," 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC), Bengaluru, India, 2022, pp. 143-145, doi: 10.1109/IIHC55949.2022.10059639.

[4] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo and J. Barata, "Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook," in IEEE Access, vol. 8, pp. 220121-220139, 2020, doi: 10.1109/ACCESS.2020.3042874.

[5] Chohan, Jasgurpreet Singh and Kumar, Raman and Kumar, Sandeep and Goyal, Bhawna and Dogra, Ayush and Kukreja, Vinay, "Implementation of Artificial Intelligence and Machine Learning in Manufacturing", 2023 2nd International Conference on Edge Computing and Applications (ICECAA), 497-503, 10.1109/ICECAA58104.2023.10212238

[6] Wen, Shuo and Wang, Peixin and Zheng, Jinying and Huang, Weiting and Ye, Zhilong and Li, Xiaomin and Zhao, Lixuan, "Sentiment Analysis of Social Media Comments based on Random Forest and Support Vector Machine", 2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), 277-281, 10.1109/AEECA55500.2022.9918957

[7] Lokesh, Degala and J, Femila Roseline, "Probability of a Device Failure using Support Vector Machine by comparing with Random Forest Algorithm to improve the accuracy", 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), 1-5, 10.1109/ICECONF57129.2023.10083595

[8] Atul and Shankar, Pranav and Rallabandi, Srikari and Singabhattu, Bhavya and Shinde, Ashish, "Evaluation of Machine Learning Approaches for Accurate Heart Disease Prediction", 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 10.1109/ICCCNT56998.2023.10306927

[9] Carratù, Marco and Gallo, Vincenzo and Iacono, Salvatore Dello and Sommella, Paolo and Bartolini, Alessandro and Grasso, Francesco and Ciani, Lorenzo and Patrizi, Gabriele, "A Novel Methodology for Unsupervised Anomaly Detection in Industrial Electrical Systems", IEEE Transactions on Instrumentation and Measurement, 72, 1-12, 10.1109/TIM.2023.3318684

[10] Teoh, Yyi Kai and Gill, Sukhpal Singh and Parlikad, Ajith Kumar, "IoT and Fog-Computing-Based Predictive Maintenance Model for Effective Asset Management in Industry 4.0 Using Machine Learning", IEEE Internet of Things Journal, 10, 3, 2087-2094, 10.1109/JIOT.2021.3050441

[11] Lima, A.L.d.C.D., Aranha, V.M., Carvalho, C.J.d.L. et al. Smart predictive maintenance for high-performance computing systems: a literature review. J Supercomput 77, 13494–13513 (2021). https://doi.org/10.1007/s11227-021-03811-7

[12] Sang, Go Muan and Xu, Lai and de Vrieze, Paul and Bai, Yuewei and Pan, Fangyu, "Predictive Maintenance in Industry 4.0, Proceedings of the 10th International Conference on Information Systems and Technologies 2021, 29, 11, 10.1145/3447568.3448537

[13] Taubleau, "Time Series Forecasting: Definition, Applications, and Examples", https://www.tableau.com/learn/articles/time-series-forecasting

[14] Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2023). Machine learning advances for time series forecasting. J Econ Surv, 37, 76–111. https://doi.org/10.1111/joes.12429

[15] Gen Li, Jason J. Jung, Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges, Information Fusion, Volume 91, 2023, Pages 93-102, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2022.10.008.

[16] R. Sharma, N. Singh and S. Birla, "An Experimental Study for Comparing Different Method for Time Series Forecasting Prediction & Anomaly Detection," 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 2021, pp. 1-4, doi: 10.1109/ICECCT52121.2021.9616662.

[17] D. Ostroški, K. Slovenec, I. Brajdić and M. Mikuc, "Anomaly Correction in Time Series Data for Improved Forecasting," 2021 16th International Conference on Telecommunications (ConTEL), Zagreb, Croatia, 2021, pp. 85-88, doi: 10.23919/ConTEL52528.2021.9495986.

[18] S. Sanyal and A. Chattopadhyay, "DeepMines: A fog Enabled Prediction Platform for Underground Coal Mines," 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, 2020, pp. 726-730, doi: 10.1109/COMSNETS48256.2020.9027454.

[19] S. B, S. R, S. G. R, S. P. R. V, R. P. H and S. R, "Tracking of Food Waste in Food Supply Chain Using Machine Learning," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-4, doi: 10.1109/ICECONF57129.2023.10083755.

[20] Kashpruk, N.; Piskor-Ignatowicz, C.; Baranowski, J. Time Series Prediction in Industry 4.0: A Comprehensive Review and Prospects for Future Advancements. Appl. Sci. 2023, 13, 12374. https://doi.org/10.3390/app132212374

[21] Ayvaz, Serkan & Alpay, Koray. (2021). Predictive Maintenance System for Production Lines in Manufacturing: A Machine Learning Approach Using IoT Data in Real-Time. Expert Systems with Applications. 173. 114598. 10.1016/j.eswa.2021.114598.

[22] Megha Sisode, Manoj Devare, "A Review on Machine Learning Techniques for Predictive Maintenance in Industry 4.0," Proceedings of the International Conference on Applications of Machine Intelli-

gence and Data Analytics (ICAMIDA 2022), Atlantis Press, 2023, https://doi.org/10.2991/978-94-6463-136-4_67
[23] P. d'Agostino, M. Violante and G. Macario, "A user-extensible solution for deploying fog computing in industrial applications," 2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE), Helsinki, Finland, 2023, pp. 1-6, doi: 10.1109/ISIE51358.2023.10227939.
[24] P. D'Agostino, M. Violante and G. Macario, "An Embedded Low-Cost Solution for a Fog Computing Device on the Internet of Things," 2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC), Tartu, Estonia, 2023, pp. 284-291, doi: 10.1109/FMEC59375.2023.10306045.
[25] Evalyne Ndanu, "Fog Computing?", African Women in Technology, 2021, https://www.africanwomenintech.com/fog-computing/
[26] M. Hunko, V. Tkachov, A. Kovalenko and H. Kuchuk, "Advantages of Fog Computing: A Comparative Analysis with Cloud Computing for Enhanced Edge Computing Capabilities," 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 2023, pp. 1-5, doi: 10.1109/KhPIWeek61412.2023.10312948