Exploratory Data Analysis for PiSA Farmacéutica

Jose Angel Govea Garcia[1], Daniel Alberto Sánchez Fortiz[1] and Diego Vértiz Padilla[1]

[1] *Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Campus Guadalajara, México*

Reception date of the manuscript: 09/03/2025
Acceptance date of the manuscript: 16/03/2025
Publication date: 16/03/2025

Abstract—This study addresses the issue of premature deterioration of a bottle-blowing machine used in the production of Electrolit serum at PiSA Pharmaceuticals. Through an exploratory data analysis (EDA) of records from over 100 sensors, anomalous fluctuations were identified in key temperature variables. An anomaly detection model based on K-Nearest Neighbors (KNN) was implemented, enabling the detection of critical events and coinciding anomalies across multiple variables. To validate these findings, a SARIMA time series forecasting model was developed, serving as a reference for the expected behavior of the analyzed variables. The comparison between the actual data and the model's predictions confirmed the hypothesis that thermal stress contributes to the accelerated wear of the machine. The results reinforce the need for predictive maintenance strategies that can anticipate failures and extend the equipment's lifespan.

Keywords—Exploratory Data Analysis, Anomaly Detection, KNN, SARIMA, Time Series, Pharmaceutical Industry, Predictive Maintenance, Statistical Models, Model Validation

I. Introduction

PiSA Farmacéutica is a Mexican company with an international presence. The factory produces more than 1,500 pharmaceutical brands across 17 specialty lines and operates 14 production plants in Mexico. Its most well-known product worldwide is Electrolit serum, which is packaged in bottles manufactured using a blow molding machine. This machine is critical to the process, as it ensures that the bottles are properly inflated.

II. Problem

The blow molding machine responsible for manufacturing bottles for Electrolit serum has experienced a significant reduction in its operational lifespan. While this machinery is expected to operate for at least one year before requiring replacement or major maintenance, it currently reaches a maximum of only two to three months of continuous operation before presenting critical failures. This situation leads to production stoppages, increases maintenance costs, and directly impacts the plant's operational efficiency and profitability.

Although there is a data log from more than 100 sensors installed on the machine, a thorough analysis has yet to be conducted to understand the behavior of the monitored variables and identify patterns that could explain or predict these failures. Given this situation, there is a need to perform an Exploratory Data Analysis (EDA), apply anomaly detection techniques, and develop predictions that enable anticipation of potential issues and improve the machine's reliability.

III. Hypothesis

The thermal stress generated during the blow molding process of Electrolit bottles is causing accelerated deterioration in the lifespan of the blow molding machine. This phenomenon is reflected in the anomalous fluctuations of the temperature variables recorded by the sensors, indicating potential failures in the system's thermal control.

IV. Methodology

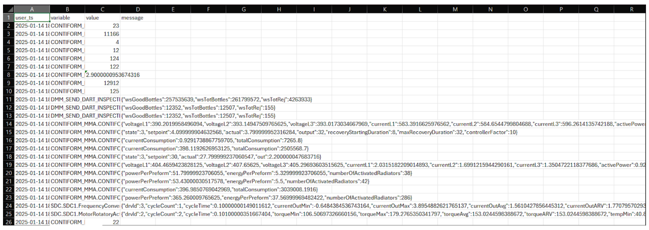
a. Data

Two files provided by PiSA were analyzed, each approximately 1.8 GB in size, corresponding to the months of November 2024 and January 2025. Each dataset contains around 45 million records in Parquet format.

b. Data Processing

The raw data from the blow molding machine's monitoring system was processed using the pandas library in Python, along with pyarrow and numpy. The source files were in Parquet format, a columnar storage format designed to efficiently handle large volumes of data. Given the size of the original files, a batch processing scheme (*chunks*) of 100,000 records was implemented. This approach allowed for controlled reading of the data, reducing memory consumption and facilitating filtering within specific time windows.

Contact data: José Ángel Govea García, goveaangel090@gmail.com



	A	B	C	D
1	user_ts	variable	value	message
2	2025-01-14 18	CONTIFORM_	23	
3	2025-01-14 18	CONTIFORM_	11166	
4	2025-01-14 18	CONTIFORM_	4	
5	2025-01-14 18	CONTIFORM_	12	
6	2025-01-14 18	CONTIFORM_	124	
7	2025-01-14 18	CONTIFORM_	122	
8	2025-01-14 18	CONTIFORM_	2.9000000953674316	
9	2025-01-14 18	CONTIFORM_	12912	
10	2025-01-14 18	CONTIFORM_	125	
11	2025-01-14 18	DMM_SEND_DART_INSPECTI		{"wsGoodBottles":257535639,"wsTotBottles":261799572,"wsTotRej":4263933}
12	2025-01-14 18	DMM_SEND_DART_INSPECTI		{"wsGoodBottles":12352,"wsTotBottles":12507,"wsTotRej":155}
13	2025-01-14 18	DMM_SEND_DART_INSPECTI		{"wsGoodBottles":12352,"wsTotBottles":12507,"wsTotRej":155}
14	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"voltageL1":390.2019958496094,"voltageL2":393.14947509765625,"voltageL3":393.0173034667969,"currentL1":583.3916625976562,"currentL2":584.6544799804688,"currentL3":596.2614135742188,"activePowe
15	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"state":3,"setpoint":4.099999904632568,"actual":3.799999952316284,"output":32,"recoveryStartingDuration":8,"maxRecoveryDuration":32,"controllerFactor":10}
16	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"currentConsumption":0.9291738867759705,"totalConsumption":7265.8}
17	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"currentConsumption":398.1192626953125,"totalConsumption":2505568.7}
18	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"state":3,"setpoint":30,"actual":27.799999237060547,"out":2.200000047683716}
19	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"voltageL1":404.4659423828125,"voltageL2":407.65625,"voltageL3":405.29693603515625,"currentL1":2.0315182209014893,"currentL2":1.6991215944290161,"currentL3":1.3504722118377686,"activePower":0.92
20	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"powerPerPreform":51.79999923706055,"energyPerPreform":5.329999923706055,"numberOfActivatedRadiators":38}
21	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"powerPerPreform":53.43000030517578,"energyPerPreform":5.5,"numberOfActivatedRadiators":42}
22	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"currentConsumption":396.9850769042969,"totalConsumption":3039008.1916}
23	2025-01-14 18	CONTIFORM_MMA.CONTIFC		{"powerPerPreform":365.260009765625,"energyPerPreform":37.56999969482422,"numberOfActivatedRadiators":286}
24	2025-01-14 18	SDC.SDC1.FrequencyConve		{"drvId":3,"cycleCount":1,"cycleTime":0.10000000149011612,"currentOutMin":-0.6484384536743164,"currentOutMax":3.8954882621765137,"currentOutAvg":1.5610427856445312,"currentOutARV":1.77079570293
25	2025-01-14 18	SDC.SDC1.MotorRotatoryAc		{"drvId":2,"cycleCount":2,"cycleTime":0.10100000351667404,"torqueMin":106.50697326660156,"torqueMax":179.2765350341797,"torqueAvg":153.0244598388672,"torqueARV":153.0244598388672,"tempMin":40.8
26	2025-01-14 18	CONTIFORM_	22	

Fig. 1: Original Dataset.

During the iteration over each chunk, records were filtered to retain only those whose timestamp (user ts) fell within the established range. The results were consolidated into a single DataFrame representing all available data within the selected time window.

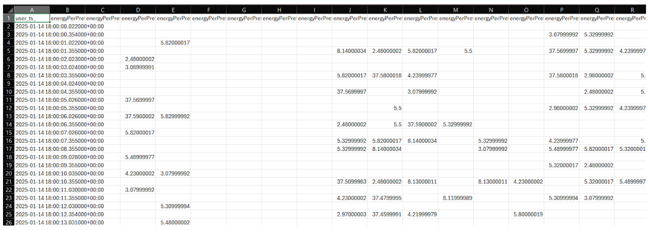
The raw dataset contained four key columns: user_ts, variable, value, and message. In particular, the message column stored additional information in JSON format, which was converted into a dictionary for subsequent expansion into additional columns.

A list of variables of interest was defined, selected based on their relevance to monitoring the thermal behavior of the machine. These included temperatures of the preform necks, cooling circuit temperatures, and readings from the heating control layers, among others.

Subsequently, a pivoting process was carried out, transforming the data structure from long format to wide format. This allowed the measurements to be grouped by timestamp (user ts) and each variable to be distributed into an independent column.

Upon observing that many of the data points were null, we aggregated each variable by second, grouping records by second instead of by millisecond. This provided better data handling for analysis as time series.

Finally, additional cleaning of the variable names was performed to facilitate interpretation, eliminating redundant prefixes such as `CONTIFORM MMA.CONTIFORM MMA1`.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	user_ts_	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPre	energyPerPr
2	2025-01-14 18:00:00.022000+00:00																	
3	2025-01-14 18:00:00.354000+00:00															3.07999992	5.32999992	
4	2025-01-14 18:00:01.022000+00:00				5.82000017													
5	2025-01-14 18:00:01.355000+00:00									8.14000034	2.48000002	5.82000017	5.5			37.5699997	5.32999992	4.2399997
6	2025-01-14 18:00:02.023000+00:00			2.48000002														
7	2025-01-14 18:00:03.024000+00:00			3.08999991														
8	2025-01-14 18:00:03.355000+00:00									5.82000017	37.5800018	4.23999977				37.5800018	2.98000002	5.
9	2025-01-14 18:00:04.024000+00:00																	
10	2025-01-14 18:00:04.355000+00:00									37.5699997		3.07999992					2.48000002	5.
11	2025-01-14 18:00:05.026000+00:00			37.5699997														
12	2025-01-14 18:00:05.355000+00:00										5.5					2.98000002	5.32999992	4.2399997
13	2025-01-14 18:00:06.026000+00:00			37.5900002	5.82999992													
14	2025-01-14 18:00:06.355000+00:00									2.48000002	5.5	37.5900002	5.32999992					
15	2025-01-14 18:00:07.026000+00:00			5.82000017														
16	2025-01-14 18:00:07.355000+00:00									5.32999992	5.82000017	8.14000034		5.32999992		4.23999977		5.
17	2025-01-14 18:00:08.355000+00:00									5.32999992	8.14000034			3.07999992		5.48999977	5.82000017	5.3200001
18	2025-01-14 18:00:09.028000+00:00			5.48999977														
19	2025-01-14 18:00:09.355000+00:00															5.32000017	2.48000002	
20	2025-01-14 18:00:10.035000+00:00			4.23000002	3.07999992													
21	2025-01-14 18:00:10.355000+00:00									37.5099983	2.48000002	8.13000011		8.13000011	4.23000002		5.32000017	5.4899997
22	2025-01-14 18:00:11.030000+00:00			3.07999992														
23	2025-01-14 18:00:11.355000+00:00									4.23000002	37.4799995		8.11999989			5.30999994	3.07999992	
24	2025-01-14 18:00:12.030000+00:00				5.30999994													
25	2025-01-14 18:00:12.354000+00:00									2.97000003	37.4599991	4.21999979			5.80000019			
26	2025-01-14 18:00:13.031000+00:00				5.48000002													

Fig. 2: Pivoted Dataset.

c. Variable Selection

To reduce noise and ensure the quality of the analysis, only those temperature variables with a sufficient amount of valid data were selected. An 80th percentile threshold was applied based on the count of available values per variable, so that only those with the greatest coverage were included in the subsequent analysis.

The resulting DataFrame included the filtered records, ready for time series analysis, in which the following conditions were ensured:

- Timestamps were unique.
- The time zone was adjusted to UTC.
- Measurements were grouped by timestamp, taking the maximum value in case of duplicates.

Some of the variables we used include:

1. `energyPerPreform_CurrentPreform NeckFinishTemperature.0`
2. `energyPerPreform_CurrentPreform TemperatureOvenInfeed.0`
3. `numberOfActivatedRadiators_CurrentPreform NeckFinishTemperature.0`
4. `powerPerPreform_CurrentPreform NeckFinishTemperature.0`
5. `value_ActualTemperatureCoolingCircuit2.0`
6. `value_CurrentPreformTemperature OvenInfeed.0`

d. Anomaly Detection

For anomaly detection in the analyzed time series, the K-Nearest Neighbors (KNN) algorithm was used, leveraging the implementation provided by the PyOD (Python Outlier Detection) library. This method is based on the premise that anomalous observations exhibit behavior significantly different from the rest of the data, which is reflected in greater distances from their nearest neighbors within the multidimensional space.

KNN was chosen due to its simplicity and effectiveness for detecting *outliers* in *datasets* with non-linear characteristics and without specific distribution assumptions. In addition, its ability to adapt to different data scales makes it a suitable option for the problem at hand.

Prior to applying the model, the time series were smoothed using a 10-observation moving average, in order to reduce the impact of extreme noise and avoid false positives in the detection process.

The KNN algorithm identifies anomalies by calculating the average distance from each point to its k nearest neighbors. In this analysis, the hyperparameters were determined as follows:

- n neighbors = 20
- contamination = 0.0025

This means that the model assumes that 0.25% of the analyzed data is anomalous. These hyperparameters were adjusted to achieve an appropriate balance between the model's sensitivity (its ability to detect true anomalies) and its precision (minimizing false positives).

After detection, a temporal coincidence analysis was performed, identifying events where multiple variables simultaneously presented anomalies, reinforcing the hypothesis of systemic failures.

The results of the analysis made it possible to identify observations whose behavior significantly deviated from the general trend of the time series. These observations were classified as anomalous and are visually highlighted in the result graphs.

e. SARIMA

After developing our anomaly detector, we sought to strengthen the analysis to move towards a solid refutation of our hypothesis. With this objective, we decided to forecast the expected behavior of the temperature variables under normal conditions. This allowed us to compare the actual behavior of the data at the moments of anomaly with the projected behavior, thus identifying significant deviations.

For this forecast, we implemented a SARIMA model (Seasonal AutoRegressive Integrated Moving Average), selected for its ability to model time series that exhibit both trends and seasonality—characteristics observed in the variables analyzed.

The process we followed to implement the SARIMA model was as follows:

1. **Target Variable Selection**: The first step was to select the variables to be modeled. The analysis focused on three variables, which presented significant findings during the anomaly detection process. These variables were:

   - `energyPerPreform_CurrentPreformNeckFinishTemperature.0`
   - `powerPerPreform_CurrentPreformNeckFinishTemperature.0`
   - `numberOfActivatedRadiators_CurrentPreformNFinishTemperature.0`

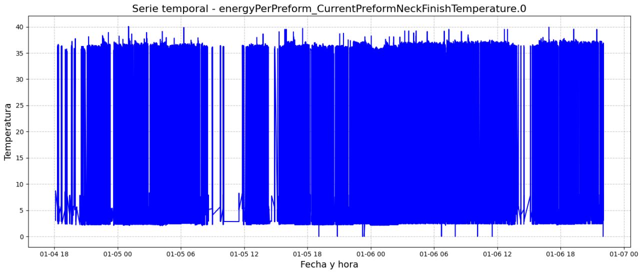


Fig. 3: Time series of the variable energyPerPreform_CurrentPreformNeckFinishTemperature.0.

2. **Preliminary Testing**: Subsequently, the time series of the selected variables were analyzed to understand their behavior. Augmented Dickey-Fuller (ADF) tests were performed to verify stationarity, and the autocorrelation function (ACF) and partial autocorrelation function (PACF) were examined to validate the feasibility of a SARIMA model.

3. **Temporal Sampling**: Due to computational limitations, it was not possible to work with the entire dataset at its original resolution (per second). Therefore, sampling at 10-second intervals was performed, and subsequently, an interpolation process was applied to approximate the actual amount of data during the prediction phase. This approach allowed for preserving the largest amount of relevant information while reducing the computational load.

4. **SARIMA Hyperparameter Definition**: The best parameters for the ARIMA component were identified through an iterative process of empirical validation. Once these were defined, the parameters for the seasonal component of the model were determined. The separation of this process was due to the previously mentioned computational constraints.

5. **SARIMA Model Training**: With the hyperparameters established, the model was trained using a *sliding window* of 7 days for training and 2 days for prediction, advancing 2 days per iteration until covering the required prediction period. Once again, the 10-second sampling was used to maintain computational efficiency.

6. **Residual Analysis**: After training, a thorough analysis of the residuals was conducted. It was confirmed that they behave mostly as white noise, with no significant autocorrelation. Additionally, homoscedasticity was observed, evidenced by an almost constant variance over time. The distribution of the residuals was also evaluated, yielding results close to normality.

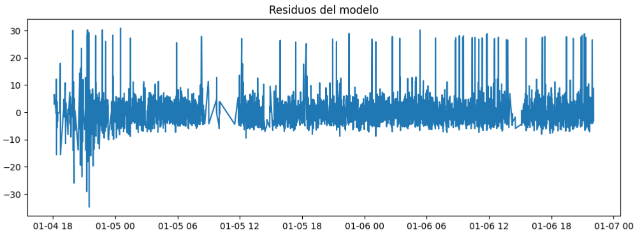


Fig. 4: Behavior of the SARIMA Model Residuals.

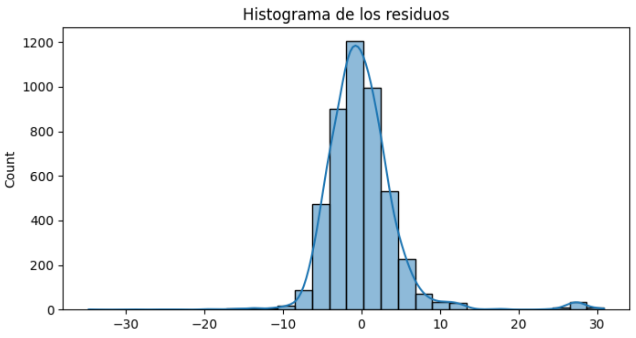


Fig. 5: Histogram of the SARIMA Model Residuals.

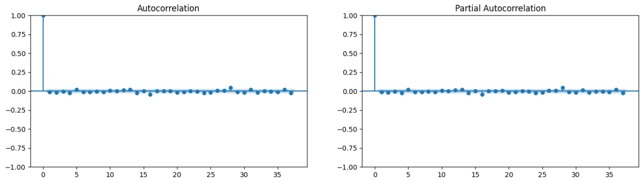


Fig. 6: Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the SARIMA Model Residuals

7. **Model Evaluation**: Performance metrics were calculated for each prediction, using MAE (*Mean Absolute Error*) and RMSE (*Root Mean Squared Error*). The results indicated an acceptable level of prediction error, reflecting the model's ability to capture the expected behavior of the time series under normal conditions.

The forecast obtained through SARIMA served as a reference model to compare with the actual observed data. In cases where the real behavior significantly deviated from the forecast, the occurrence of anomalies previously detected by the KNN algorithm was confirmed, further reinforcing the hypothesis of failures or abnormal behaviors within the system.

The complete implementation code for the SARIMA model, as well as the data processing and anomaly detection using KNN, is available in the following repository:

V. Results

The results are divided into two main phases: anomaly detection using the K-Nearest Neighbors (KNN) algorithm, and subsequent validation through SARIMA forecasting.

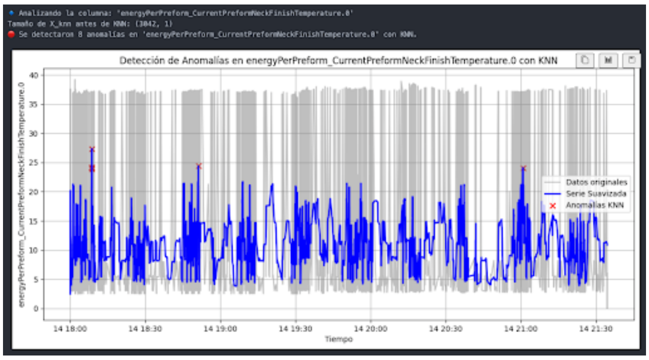
a. Anomaly Detection - KNN

The analysis revealed the presence of multiple anomalies throughout the time series. However, most of these corresponded to singular events, meaning anomalies that appeared in isolation within a single variable, making interpretation and the identification of a significant pattern more difficult.

The most relevant finding emerged when it was identified that three specific variables presented anomalies simultaneously, coinciding exactly at the same second on multiple occasions. These variables were:

- `energyPerPreform_CurrentPreformNeckFinish Temperature.0`
- `powerPerPreform_CurrentPreformNeckFinish Temperature.0`
- `numberOfActivatedRadiators_CurrentPreformNeck FinishTemperature.0`

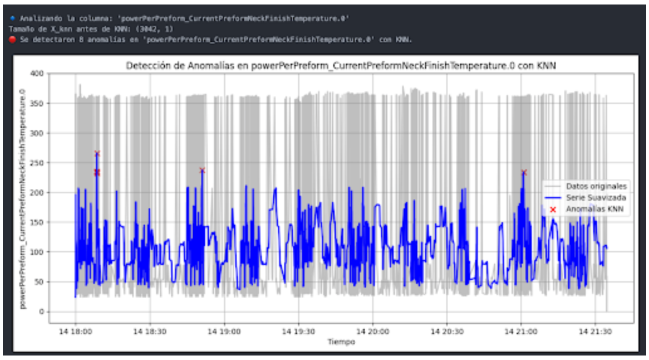
The joint occurrence of anomalies in these three variables suggests the existence of abnormal events related to machine malfunction or operating conditions outside the established parameters. Based on this finding, it was decided to focus the analysis exclusively on these three variables.



```
🔹 Analizando la columna: 'energyPerPreform_CurrentPreformNeckFinishTemperature.0'
Tamaño de X_knn antes de KNN: (3042, 1)
🔴 Se detectaron 8 anomalías en 'energyPerPreform_CurrentPreformNeckFinishTemperature.0' con KNN.
```

Detección de Anomalías en energyPerPreform_CurrentPreformNeckFinishTemperature.0 con KNN
energyPerPreform_CurrentPreformNeckFinishTemperature.0
0
5
10
15
20
25
30
35
40
14 18:00
14 18:30
14 19:00
14 19:30
14 20:00
14 20:30
14 21:00
14 21:30
Tiempo
Datos originales
Serie Suavizada
Anomalías KNN

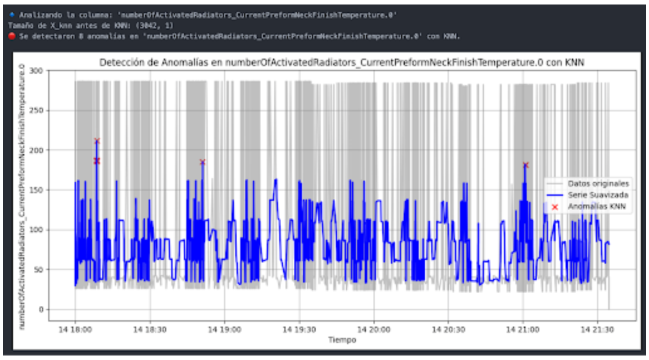
Fig. 7: Anomaly Detection of the variable energyPerPreform_CurrentPreformNeckFinish Temperature.0



```
🔹 Analizando la columna: 'powerPerPreform_CurrentPreformNeckFinishTemperature.0'
Tamaño de X_knn antes de KNN: (3042, 1)
🔴 Se detectaron 8 anomalías en 'powerPerPreform_CurrentPreformNeckFinishTemperature.0' con KNN.
```

Detección de Anomalías en powerPerPreform_CurrentPreformNeckFinishTemperature.0 con KNN
powerPerPreform_CurrentPreformNeckFinishTemperature.0
0
50
100
150
200
250
300
350
400
14 18:00
14 18:30
14 19:00
14 19:30
14 20:00
14 20:30
14 21:00
14 21:30
Tiempo
Datos originales
Serie Suavizada
Anomalías KNN

Fig. 8: Anomaly Detection of the variable powerPerPreform_CurrentPreformNeckFinish Temperature.0



```
🔹 Analizando la columna: 'numberOfActivatedRadiators_CurrentPreformNeckFinishTemperature.0'
Tamaño de X_knn antes de KNN: (3042, 1)
🔴 Se detectaron 8 anomalías en 'numberOfActivatedRadiators_CurrentPreformNeckFinishTemperature.0' con KNN.
```

Detección de Anomalías en numberOfActivatedRadiators_CurrentPreformNeckFinishTemperature.0 con KNN
numberOfActivatedRadiators_CurrentPreformNeckFinishTemperature.0
0
50
100
150
200
250
300
14 18:00
14 18:30
14 19:00
14 19:30
14 20:00
14 20:30
14 21:00
14 21:30
Tiempo
Datos originales
Serie Suavizada
Anomalías KNN

Fig. 9: Anomaly Detection of the variable numberOfActivatedRadiators_CurrentPreformNeck FinishTemperature.0

b. SARIMA

The results obtained through the SARIMA model reflect solid performance in forecasting the three selected variables. The model effectively captured the trends and seasonality present in the time series, as evidenced by the consistency of the overall forecast and the error metrics (MAE and RMSE), which remained within acceptable ranges during the different iterations of the sliding window. This indicates that, under normal operating conditions, the model is capable of predicting the expected behavior of the analyzed variables with an adequate level of accuracy.

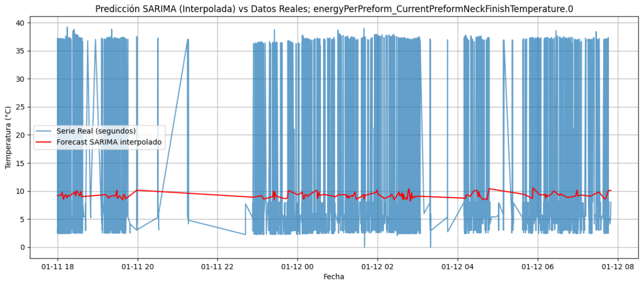


Fig. 10: SARIMA model forecast compared to the actual values of the variable energyPerPreform_CurrentPreformNeckFinishTemperature.0.

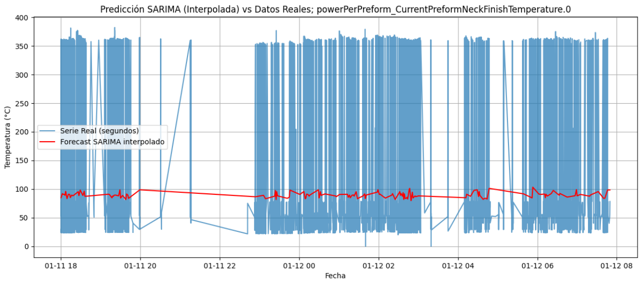


Fig. 11: SARIMA model forecast compared to the actual values of the variable powerPerPreform_CurrentPreformNeckFinishTemperature.0.

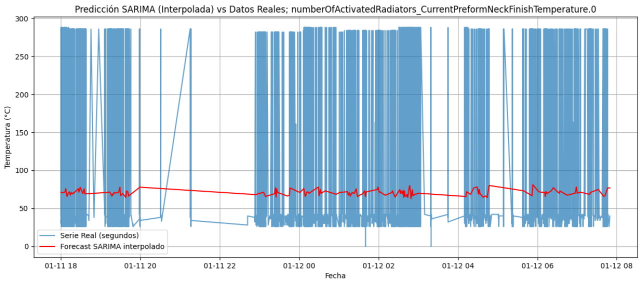


Fig. 12: SARIMA model forecast compared to the actual values of the variable numberOfActivatedRadiators_CurrentPreformNeckFinishTemperature.0.

However, it is important to note that due to the sampling process used to reduce the data granularity (from records per millisecond/second to records every 10 seconds), the model generates a smoothed forecast. This smoothing may limit the model's sensitivity to capture abrupt events or rapid fluctuations in the signal, making it difficult to confirm with absolute certainty the magnitude and exact timing of certain anomalies previously detected by the KNN algorithm.

Despite these limitations, the SARIMA model fulfilled its main purpose: providing a reliable reference of the normal behavior of the variables, allowing for a comparison with the actual data and highlighting significant deviations during periods where group anomalies were detected. This reinforces the proposed hypothesis regarding the accelerated deterioration of the machine and lays the foundation for the development of a more robust predictive maintenance system in the future.

VI. Conclusions

These findings confirm the proposed hypothesis: anomalous fluctuations in the temperatures recorded by the sensors generate additional thermal stress on the blow molding machine, contributing to the accelerated deterioration of its components. The combination of anomaly detection methods and prediction models not only validates this hypothesis but also provides a solid foundation for implementing predictive maintenance strategies aimed at extending the equipment's lifespan and reducing downtime.