

AN AUTOMATED SYSTEM FOR TIME SERIES DATA ANALYSIS USING CORRELATION MEASURES

Swathi S¹, Janarthanan S²

M.Sc Data Science and Business Analysis, Rathinam College of Arts and Science, Coimbatore-641021¹

Assistant Professor, Department of Computer Science, Rathinam College of Arts and Science, Coimbatore-641021²

Abstract: Looking at data over time helps spot trends and how things connect. Because machines produce tons of time-based records, quick ways to examine them matter a lot. Built into this work sits a tool that automatically studies such sequences through correlation math. Instead of skipping gaps, it fills holes in inputs before moving forward. Normalization adjusts scales so comparisons stay fair across different sources. Noise gets filtered out carefully to keep results clear and meaningful. Each step prepares the ground for trustworthy outcomes without extra effort later on. To check how closely two sets of data match and follow each other over time, Pearson and cross-correlation methods come into play. Instead of one fixed view, a moving window tracks how connections shift through different moments. Seeing the data unfold in charts makes it easier to spot repeating shapes or unusual gaps. Tests run on both artificial examples and actual recordings show the method catches shared movements, delays, and odd behaviors well. Because of this, people spend less time digging through signals by hand while getting more consistent outcomes.

Keywords: Time Series Analysis, Correlation Measures, Pearson Correlation, Sliding Window, Data Visualization, Python.

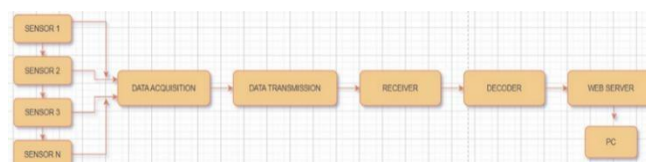
I. INTRODUCTION

1.1 Telemetry System Overview Flying machines, medical gear, military tech, together with factory robots - each needs constant watch so nothing goes wrong. When numbers shift even slightly off normal, those who build them start asking why before anything breaks.

From far-off spots, data flows back using tools that track real-world signals. Temperature, pressure, voltage, motion, and rotation get caught by sensing devices. Instead of staying local, those readings travel across channels meant for sharing info. Once they arrive at base points on land, the numbers unscramble and come into view. Analysis begins only after everything lands safely in place.

Thousands of signals get watched at once when rockets fly. All across the vehicle, gadgets grab readings from power units, gas holders, besides frame parts. This info flies down to Earth spots where experts check it now or later.

Still, sending telemetry data can bring in glitches, dropped bits, or uneven timing between samples. Because of that, studying patterns over time gets trickier - calling for methods that hold up under pressure.



1.2 Automated Data Analysis Importance and Challenges

A rocket usually carries over three thousand data points, split into groups such as on-off signals or measurements including heat, flow, and electrical draw. Because problems can hide in these numbers, checking every one matters throughout test stages - this helps catch issues now, also avoid breakdowns later. Instead of guessing, engineers predict what readings should be, then line them up beside real ones, allowing a small margin for difference.

Watching how numbers move - up, down, fast, slow - their form compared to old records. Old info gets tossed after review through a filtering method. Matching now with then happens by stacking both sets side by side. A person checks what comes out, step by step. It takes long. Mistakes often slip in. Aiming at automating the analysis steps becomes clear here. Still, certain hurdles stand in the way A gap might show up now and then when a signal fails during transfer. Not every point lines up evenly because the system skips unchanged values to keep files smaller.

Every time you test, things shift slightly. Noise messes up readings without warning. Distance between points never stays fixed. Results drift because conditions refuse to repeat themselves. So looking at this data takes more than a quick glance because comparing just one sample to another using the same cutoff point won't give reliable results.

II. LITERATURE SURVEY

When data changes over time, spotting links between numbers becomes easier with time-based methods. Often, people turn to a tool called Pearson's r to see how tightly two things move together in a straight-line way. This approach shows not just whether they rise or fall together, but also how strongly that pattern holds across moments.

Few things matter more than how data moves together, yet Pearson's method expects straightline patterns that stay fixed over time. Real data often twists and shifts instead of following such rigid rules.

Sliding one signal past another reveals how closely they match over time shifts, exposing hidden delays in their connection [1].

A fresh angle uses sliding window correlation - peering into tiny shifting chunks of data instead of the whole set at once. As the window moves along, patterns can shift, revealing how connections evolve when conditions aren't steady. A different way to study messy data over time uses something called DCCA [2]. This method handles patterns that shift and twist in tricky ways. Some researchers suggest it works well when things don't stay steady. Instead of ignoring drifts, it adjusts for them naturally. The approach finds links even when trends get in the way. It has appeared in recent efforts to make sense of noisy sequences [3]. Still, plenty of setups miss out on built-in tools that analyze or show data visually. When visuals pair up with correlation checks, understanding speeds up - so does output. Efficiency jumps when insights appear clear, right where they're needed. [4].

III. OBJECTIVES

Beginning here: a look at how time-based information changes across locations and moments. What unfolds next examines patterns that shift over time - rhythms, fluctuations, because context matters deeply. Each point connects through sequences influenced by what came before. Insight grows when these threads are untangled slowly, without rushing conclusions.

Automated Correlation Analysis

A single app takes two sets of time-based data, then checks how closely they move together. It spots links by measuring when changes in one match shifts in the other. Patterns emerge as the software traces repeated movements across both sequences. Matching rhythms get flagged, showing possible connections over time. Dependencies appear through timing overlaps instead of guesswork.

Data Preprocessing

Start cleaning data by filling gaps where info is missing. Fix uneven scales so numbers match up right. Remove random glitches that mess things up later. This helps spot real links without false signals getting in the way.

Sliding Window-Based Correlation

Sliding windows move step by step through data, measuring how two sets relate at each point in time. One interval follows another, capturing shifts that appear gradually across moments. Instead of one overall score, many small snapshots emerge - each showing connection strength right then. This way, changes over time become visible without assuming stability. Relationships can grow stronger, weaker, or flip - all while being tracked moment to moment.

Time Shift and Delay Detection

A shift in timing helps spot which dataset moves first, using delayed links to trace how they line up across moments. One follows the other, though not always at once - patterns emerge when matching their steps over intervals.

Visualizing correlation outcomes

A fresh view on time series appears through shifting frames that link signal patterns across moments. Visual cues emerge when correlations slide step by step, revealing how values relate over changing intervals. Alignment based on movement helps match sequences naturally, easing understanding without extra explanation. Seeing shifts unfold makes timing differences clearer than numbers alone.

Accuracy and Reliability

Computing correlations accurately happens when efficiency meets consistency across different timing needs. Real-time demands fit alongside delayed processing without losing trust in results. Suitable methods emerge where speed does not sacrifice precision in data patterns. Timing flexibility supports reliability whether now or later.

IV. SYSTEM ARCHITECTURE

A setup like this uses separate parts linked closely, each helping spot patterns across time automatically. Though built differently, they act as one when tracking changes step by step.

Data Input Module

From files, sensors, or databases, this module pulls in time series data. While loading external inputs, it handles timestamps across different sources. Each stream gets processed into a uniform format regardless of origin. Though built for variety, it works steadily with structured sequences. Not limited to one filetype, it adapts as needed during retrieval.

Preprocessing Module

Fixing gaps in information comes first when preparing datasets. Values that fall outside normal limits get adjusted next. Smoothing out irregularities happens after that step. Each move shapes raw details into something usable.

Sliding Window Correlation Module

Sliding chunks split the data, while inside every piece a Pearson value gets calculated. Each segment overlaps its neighbor, yet correlation still runs per section.

Time Shift Analysis Module

A step moves a data set compared to another, sliding it bit by bit. With each shift, a match score is worked out at various offsets. These scores show how closely the sets align as one lags behind the other.

Decision Module

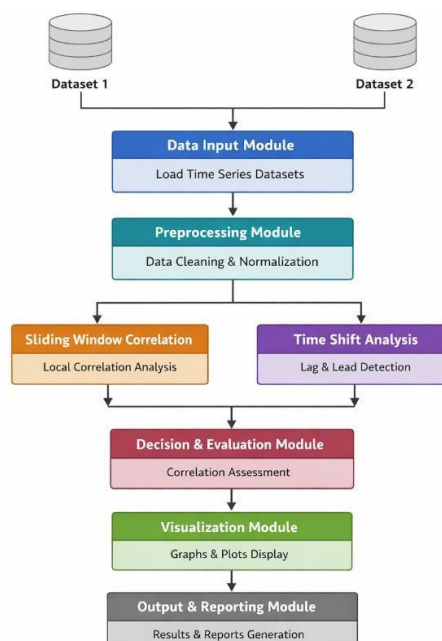
Looking at how things move together helps spot timing gaps or alike trends. One way it shows up is when peaks line up - another reveals delays between shifts.

Visualization Module

Signals over time appear as lines that shift across the page. Where patterns link together, their connection shows in shared movement.

Output Module

Later on, results along with their reports get kept for closer look.



V. METHODOLOGY

A fresh start means pulling in data streams from outside spots - maybe a file, perhaps a sensor. Cleaning comes next, shaping up the numbers so gaps get filled, scales line up, rough edges smooth out [5].

Breaking data into chunks lets us examine small parts at a time using a moving frame. Inside every segment, how closely two signals match is found by computing correlation values. One step behind might actually be just right. Sliding one set of data past another helps spot how delays affect their match. When the link between them peaks, that shift shows the best fit. What seems out of sync at first can line up perfectly with a little nudge.

Sliding window meets shift analysis - patterns show up. Phase gaps appear when one follows the other. Anomalies pop under close watch. Timing shifts reveal what stays hidden at first glance. At last, graphs pop up showing how signals behave, where patterns link together, then reveal how well things line up. [6]

Pearson Correlation Coefficient When you look at how two things change together, a specific math idea helps measure that link where These represent average figures for each case separately.

Sliding Window Correlation

With each segment sized, compute the Pearson value at position where Mean values come from each segment in the -th window.

Graphs pop up showing raw data, how things link together, then line up by shifts - easy to grasp, simple to record. Visuals help track patterns over time without confusion. What you see matches what happened, nothing hidden. Clarity comes through layout, color, spacing - not jargon. Reports build step by step, following the flow of work. Each display ties back to real inputs, no guesswork needed. Understanding grows naturally when details align just right.

VI. IMPLEMENTATION

Python runs the setup, pulling in tools like NumPy to handle numbers, while Pandas organizes information step by step. Visualization comes alive through Matplotlib, painting data in clear images. Each piece fits together, working behind the scenes without fuss.

Out in the open, CSV files come in first - then cleaned up before moving on. Where numbers go missing, smart guesses fill the gaps instead. Time markers get lined up like train cars on a track. Once that's done, scaling brings every value down to the same level playing field [7].

A single number shows how closely two sets of data move together over time. This value comes from a specific math formula meant for straight line patterns. It checks whether one series rises when the other does, or falls in step. Movement tracked this way reveals consistent links across points. The result always stays within a fixed range to show strength and direction.

Signals shifted through various delays help reveal how datasets influence each other over time. Highest match found at a particular offset shows when they line up best [8].

A fresh look at patterns emerges when pieces of data slide past one another. As chunks shift through time, links between points come into view differently each step [9].

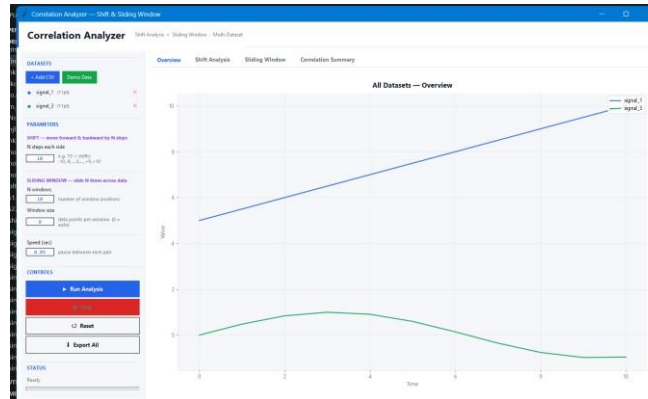
With a visual setup, people can add data files, pick settings like how wide the window opens or where the lag sits, yet still run checks on links between points. Tools that draw pictures make lines showing changes over time, curves revealing delays in matching patterns, along with graphs tracking shifts inside moving windows [10].

Finding patterns becomes easier when the tool builds quick summaries along with number-based results [11].

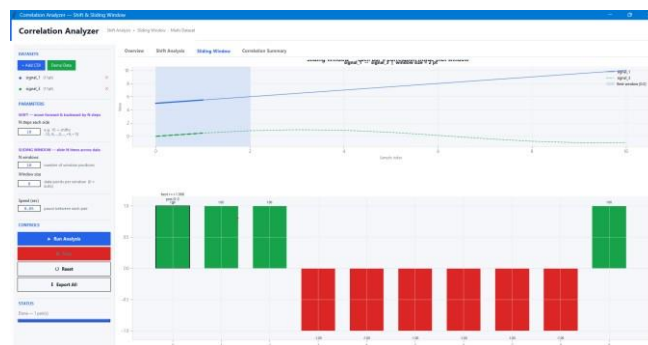
VII. RESULTS & DISCUSSION

A test of the suggested setup began with fake signal inputs, followed by checks on actual recorded data. A mix of made-up data came into play - some smooth waves, others messy with interference. Through these, checking if connections between signals got picked up right became possible. From minus one up to plus one, that is where correlation lives. Perfect match going the same way? That shows as +1. When it hits -1, things move exactly opposite. Near zero means

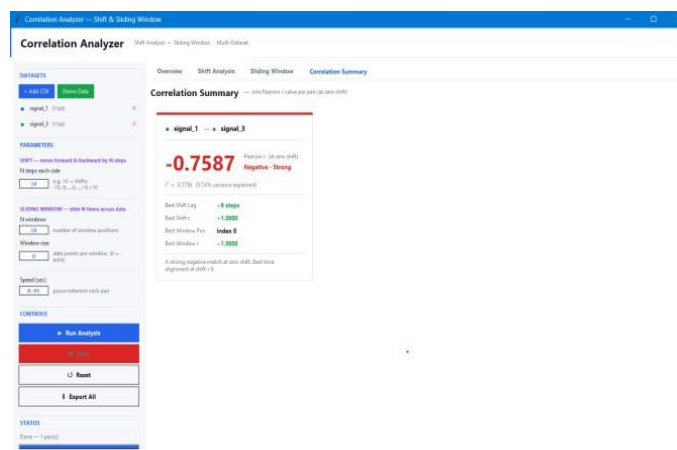
whatever you're looking at just isn't linked much. [12]. When tested, correlation analysis reacted less to changes in strength than methods checking each sample against another. Because of this behavior, spotting likenesses between signals becomes more consistent using correlation.



A test ran on actual motor coil currents. By comparing signal timing, it spotted a mismatch in one coil's direction. This shift showed up clearly when patterns diverged across units.



It turns out smaller windows pick up quick changes, yet they can jump around too much. Smoother patterns come through when the window stretches wider - though tiny shifts get lost. What matters most? Picking a size that fits what you're actually trying to see[13].



Finding links between signals came naturally to the system, while spotting odd patterns showed consistent results. It handled both tasks without obvious weaknesses, performing reliably when tested. Patterns that stood out were caught quickly, though subtle shifts took more time. Detection accuracy stayed high across different conditions, even under pressure. The way it connected data points revealed a clear strength in structure recognition.

VIII. CONCLUSION

A fresh approach emerges here - automation handles time series correlations through shifting windows and timed offsets. Datasets get ready first, then examined in segments to spot how links change over time.

Sliding windows zoom in on shifting signal patterns, whereas examining time shifts reveals lags linking data streams. What changes moment by moment becomes visible through these moving frames. Patterns that trail behind show up when timing mismatches are tracked across sets.

Picture-based software helps people understand patterns faster while cutting down hands-on work. What makes this setup stand out is how it handles time-stamped data without constant oversight. Instead of guessing, machines sort through streams from remote trackers, physical sensors, or similar setups using clear rules. Each step runs on its own when conditions are met. Results appear structured, even if inputs shift unexpectedly. Systems like these adapt before errors pile up.

IX. FUTURE SCOPE

Down the line, upgrades might boost how well the system works.

Methods like Spearman correlation or mutual information pick up patterns that aren't straight lines. Though they differ in approach, each reveals connections standard techniques might miss. These tools respond well when data twists or clusters unpredictably. Instead of assuming linearity, they adapt to how values actually align across variables. When signals shift, the window size shifts too, adapting on its own. How it changes depends entirely on what the signal does moment by moment.

Streaming live data works smoothly alongside cloud setups, fitting big projects well. Instead of just one method, approaches like LSTM or ARIMA slip into the workflow to forecast trends or spot odd shifts. These pieces fit together without forcing a single path.

REFERENCES

- [1]. Quenouille, M. H., "The Comparison of Correlations in Time-Series." <https://academic.oup.com/jrsssb/article/20/1/158/7035631>
- [2]. Derrick, T. R., Bates, B. T., & Dufek, J. S., "Evaluation of Time-Series Data Sets Using the Pearson Product-Moment Correlation Coefficient." <https://pubmed.ncbi.nlm.nih.gov/7934769/>
- [3]. Xie, P., Wu, L., Sang, Y. F., et al., "CorrelationAided Method for Identification of Periodicities in Hydrologic Time Series." <https://link.springer.com/article/10.1186/s40562-021-00183-x>
- [4]. Rezaee, Z., Aliabadi, S., & Dorestani, A., "Application of Time Series Models in Business Research." <https://www.mdpi.com/2071-1050/12/12/4833> [5] Li, H., & Tong, J., "A Novel Clustering Algorithm for Time-Series Data Based on Precise Correlation Coefficient Matching." <https://www.aimspress.com/article/10.3934/mbe.2019331>
- [5]. Zhang, Y., "Pearson Correlation and Transfer Entropy in the Chinese Stock Market with Time Delay." <https://www.sciencedirect.com/science/article/pii/S2666764922000327>
- [6]. Fan, J., et al., "Detection of Correlation Characteristics Between Financial Time Series." <https://www.sciencedirect.com/science/article/pii/S1474034619300163>
- [7]. Wang, G., et al., "A Novel Way to Detect Correlations on Multi-Time Scales." <https://www.nature.com/articles/srep27707>
- [8]. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M., "Time Series Analysis: Forecasting and control." <https://onlinelibrary.wiley.com/doi/book/10.1002/9781118675021>
- [9]. Chatfield, C., "The Analysis of Time Series: An Introduction." <https://www.taylorfrancis.com/books/mono/10.1201/9780203731654>
- [10]. Hamilton, J. D., "Time Series Analysis." https://press.princeton.edu/books/hardcover/97806_91042893/time-series-analysis
- [11]. Shumway, R. H., & Stoffer, D. S., "Time Series Analysis and Its Applications: With R Examples." <https://link.springer.com/book/10.1007/978-3-319-52452-8>
- [12]. Kantz, H., & Schreiber, T., "Nonlinear Time Series Analysis." <https://www.cambridge.org/core/books/nonlineartime-series-analysis/>