CLOCKWORKS
ANALYTICS

BUILDING ANALYTICS COMPARISON GUIDE

BMS ALARMS

versus

FAULT DETECTION

versus

FAULT DETECTION + DIAGNOSTICS

This paper was written in collaboration between
Clockworks Analytics and Nexus Labs.

Dennis Krieger is Director of Partner Success at Clockworks Analytics. Dennis supports facility service companies using Clockworks to maintain building portfolios at peak performance.

James Dice, PE, CEM, CMVP is the founder of Nexus Labs, a weekly blog, podcast, online school, and consultancy striving to create smarter, more efficient buildings. James has over 10 years of experience implementing many types of analytics and FDD software as part of energy efficiency and commissioning projects across the US.

DON'T IGNORE THE 2ND FDD

In the facilities industry, we've come to a rare consensus: operations and maintenance (O&M) teams need to stop being reactive and start being proactive. To put it into our common idioms, we want to stop "putting out fires" and start...

Well, actually, we don't exactly have an idiom for being proactive, do we? We just know we want to go to the promised land of proactivity and each fire drill reminds us that we have not yet arrived. But why haven't we arrived? It's not out of a lack of desire. According to the US Department of Energy's Operations & Maintenance Best Practices Guide, this journey to the promised land could result in a reduction in maintenance costs of 25% to 30%, a 70% to 75% elimination of system breakdowns, and a 35% to 45% reduction in equipment downtime at a 10x return on investment. If every building owner had an easy button to produce those kinds of results, they would press it incessantly.

Impacts of Proactive Maintenance

Reduction in maintenance costs	25-30%
Elimination of breakdowns	70-75%
Reduction in downtime	35-45%
Increase in production	20-25%
Return on Investment	10X

Herein lies the problem: It's not easy—and that's because we're using the wrong tools. This paper examines and compares the range of tools available to O&M teams, from building automation system (BAS) alarms to fault detection to fault detection and diagnostics (FDD).

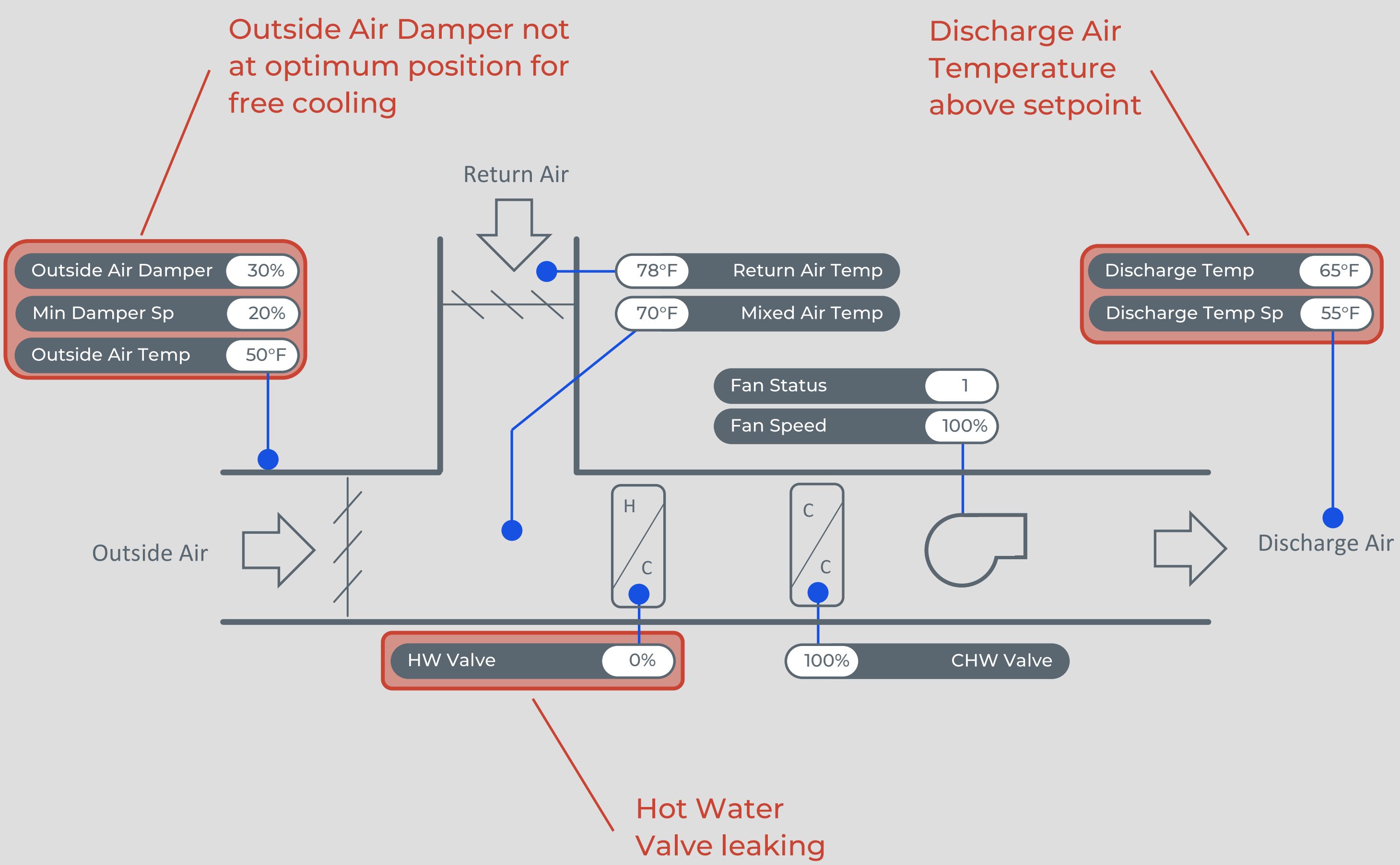
Alarms are most familiar—they come with every BAS. You get a popup that says, 'Hot Space Temp' with a nice blinking red light on the front end. Frankly, they're not very effective at supporting the move from reactive to proactive. They provide very little actionable information and are usually only slightly less reactive than using no tool at all. Worst case, they get ignored. We've all seen the BAS with thousands of unacknowledged alarms. Even if you've done a lot of work to standardize alarms and deploy them to their fullest potential, the best case is that they're a solid trigger for critical response and monitoring critical equipment.

Making the jump to an optimized building with a proactive staff requires getting ahead of critical problems and going beyond them. The tool to help with that jump is analytics software with FDD capabilities. I've helped facilities teams make this jump and have a message to report back: not all FDD tools are created equal. There are actually two kinds of FDD software: those that stop at the first "D" (Detection) and those that go all the way to Diagnostics.

This paper tells the story of that second "D" and why it's so important. To illustrate the importance, we'll use an example that every facilities team knows well: a large air handling unit, AHU_03, experiencing several issues.

The paper concludes by issuing a challenge to the industry: building owners need FDD, not just FD. If you're in the position of specifying or buying analytics software, you need to understand this distinction. Don't ignore it, or you're going to pay for it later.

How would Alarms, FD, and FDD identify, diagnose, and help you act on these issues?



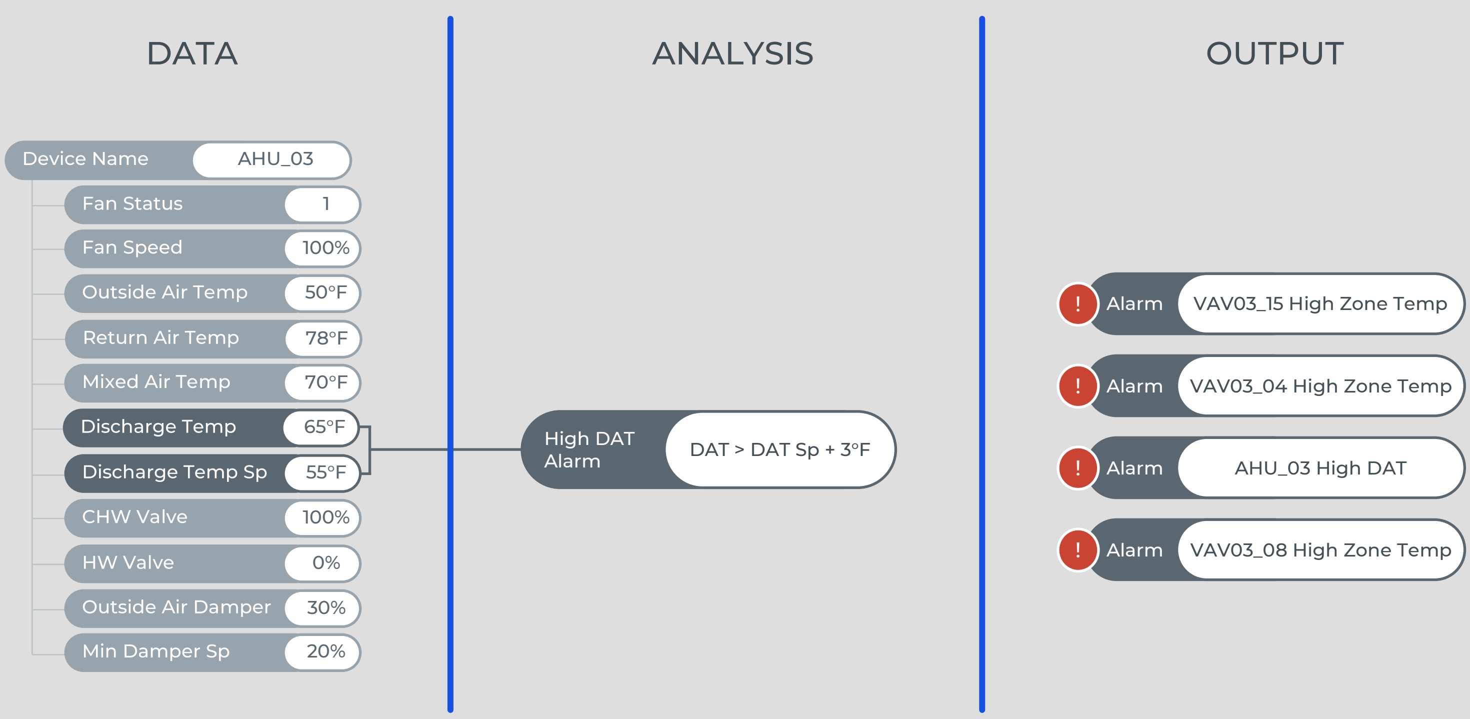
ALARMS

To trigger an alarm notification, the BAS uses the data available in the local controller to perform very simple math. Is this point greater than that point? Does the fan motor's status match the command? Has this point crossed a predefined threshold?

When it comes to making the transition from reactive to proactive, the strategic use of alarms is an improvement over using no tools at all. Alarms can be very effective at catching problems right before something bad happens and triggering a response to avoiding it. For example, a "high space temp" alarm might allow the O&M team to troubleshoot the problem before the occupant arrives on site in the morning. Since alarms come as a standard feature on every BAS, they are essentially free to use.

Frankly, despite these benefits, alarms don't help pull O&M teams out of reactive mode. To illustrate the reasons why, let's take a look at an alarm condition for our example AHU_03.

What do the BMS Alarms tell us about the AHU issues?



Three variable air volume (VAV) boxes have "High Zone Temp" alarms because the Zone Temperatures are higher than the Zone Temperature Setpoints. The "High Discharge Air Temp" alarm has been triggered because the AHU is on and the Discharge Temperature is higher than the Discharge Temperature Setpoint. The team can now begin to investigate what's going on.

Once the investigation begins, the primary problems with alarms become very clear. First, they only tell you when a problem has already occurred—meaning they won't be much help in moving from reactive to proactive. Second, they don't provide much actionable information, thus requiring a lot of work to determine what to do about the problem. How severe is it? Is this the highest priority issue on my plate right now? Is it causing comfort issues downstream? How long has this been going on? What's the root cause and how do I fix it?

When you expect an already busy team to do this much work for every issue in the building, alarms tend to pile up, overwhelm the team, and then get ignored. That's the unfortunate (yet understandable) truth.

One reason for the information deficit inherent to alarms is the poor use of data. The data in the BAS could be used to produce valuable information for operators, but it isn't. That's because an alarm's analysis capabilities are limited to simple math on only the data available in the local controller, which usually covers a short duration (a few days) and a few critical points.

In a modern building producing thousands of data points every few minutes, it's clear that more analytical horsepower would be useful. That's where analytics software comes in. But before we move on, there is one final bone to pick with alarms: they don't scale up for large portfolios.

Most portfolios of buildings have multiple BAS systems. Even if the team does an amazing job configuring the perfect alarms for a building, those alarms are still trapped within that one building, making alarm management and standardization across the portfolio an arduous task. Even if alarms are managed only at the building level, the portfolio level insights are still unavailable to the director of facilities or energy manager who has portfolio-level concerns and responsibilities.

Alarms are best when considered a last line of critical defense, but getting proactive means adding data analytics as your first line. Adding an FDD tool to the team is a great idea. Let's proceed.

FAULT DETECTION

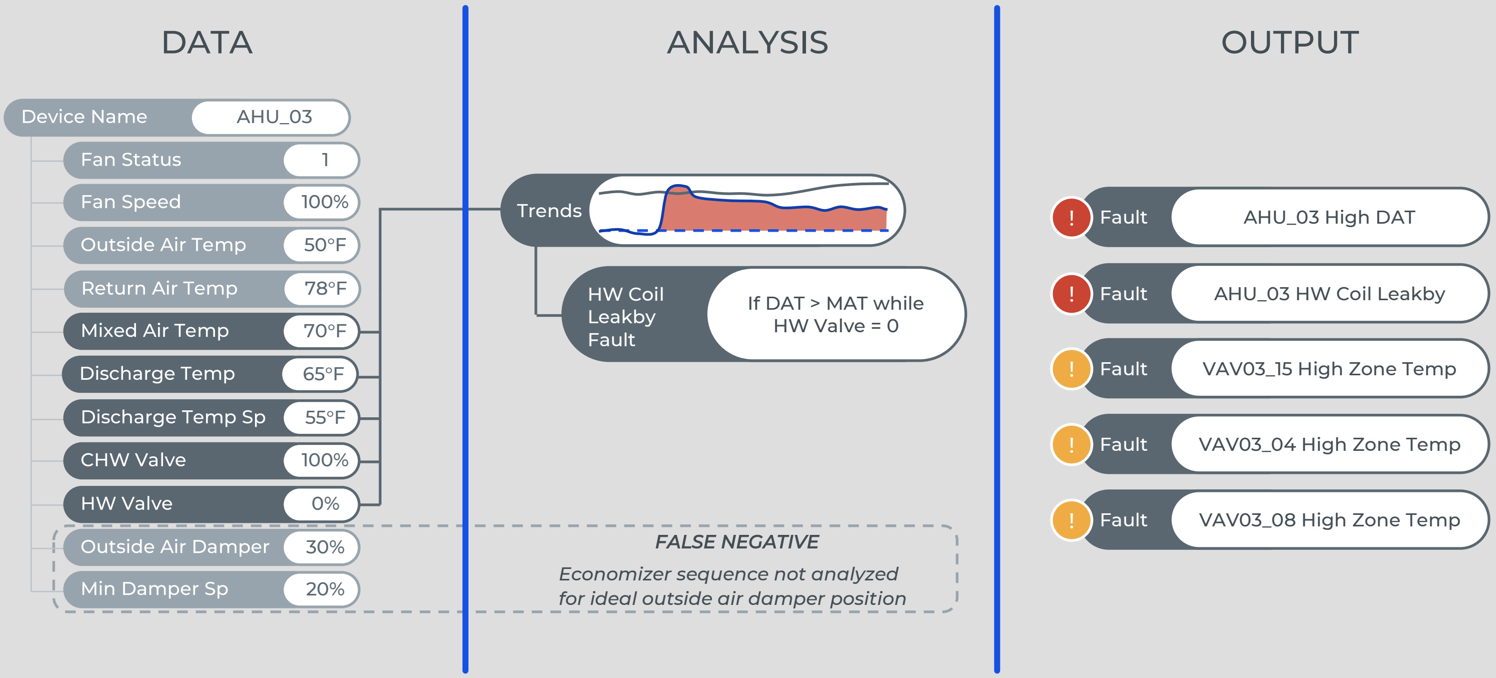
Compared to alarms, FD results in higher quality insights for O&M teams. Analytics tools provide an overlay to the existing BAS, allowing the massive amount of data to be analyzed using modern computing. Instead of just the limited data in the controller, FD rules typically include:

- Historical trend data
- Data from other systems or outside the building (e.g. local weather data)
- More advanced if-this-then-that logic rules

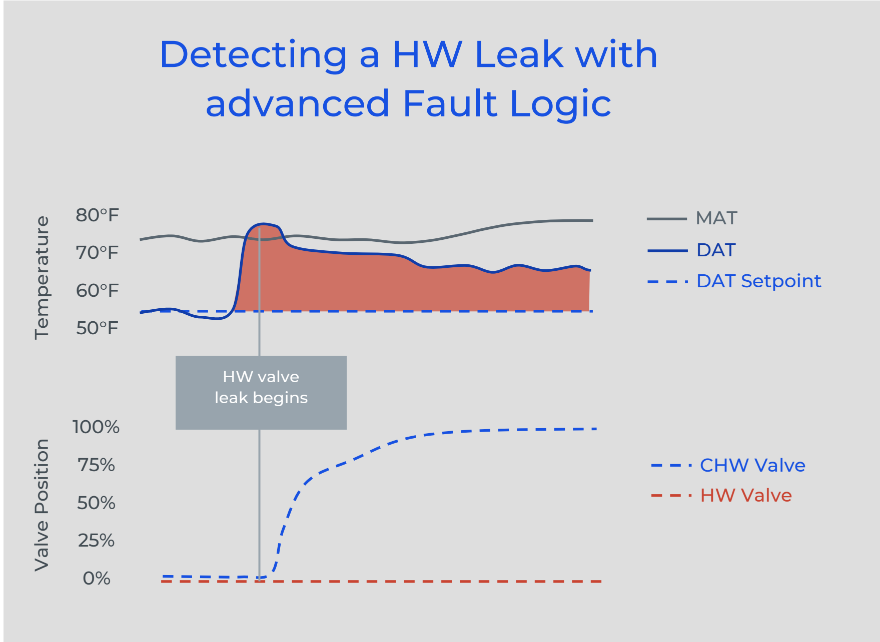
The data is used to produce specific fault notifications, such as optimization opportunities (e.g. better sequences, setpoints, and schedules) and physical components that need to be fixed (e.g. a stuck damper/leaking valve/uncalibrated sensor). These faults are filtered using some sort of ranking system based on simple importance factors, the duration the fault is active, and sometimes rule-of-thumb estimates on energy and cost savings. The software then provides data visualization tools to investigate each fault further by drilling down into data visualizations and KPIs.

To illustrate the value of the fault detection overlay software, let's review the faults that were detected on our example AHU_03:

What does Fault Detection tell us about the AHU issues?



The more advanced Fault Detection logic was able to detect the leaking hot water valve because historical data showed that the Discharge Air Temperature rapidly rose above the Mixed Air Temperature while the HW Valve was closed.



These types of insights can add value to the O&M team by surfacing problems that were previously invisible or undetected by the busy human eye. Some of these issues will have overlap with the issues detected by BAS alarms, but others will be much more proactive and predictive. The team can use the software to dig deeper into each issue to determine how long it's been going on, how severe it is, and whether it's happening in other parts of the building. If this investigation is done early enough, perhaps they could start to mitigate issues before they become bigger problems.

However, if I know one thing about O&M teams, it's that they're not in short supply of problems to fix. Does it really help them to provide more problems to solve? In my experience, analytics tools that stop at the first D add more items to the to-do list. They don't simplify. They don't take items off the to-do list. And they don't support a full move from reactive to proactive.

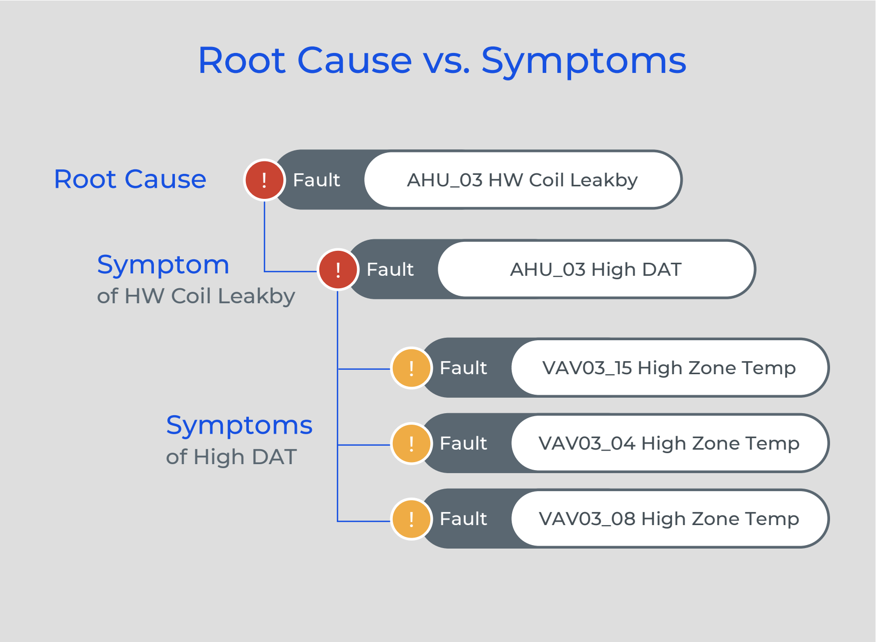
That's because when it comes to prioritizing faults and then determining the root cause of the highest priorities, the team still has a lot of work to do. Why is that? I've seen a lot of different FD tools and I've identified some fatal flaws. Let's take a deeper look at each.

The Single Fault Assumption

The first fatal flaw is what has been identified in other industries as the single fault assumption. In large complex operations (such as the control centers for refineries, or for network management), there are usually multiple outstanding problems. While the probability of simultaneous failure is slim, the probability is high that there hasn't been time to fix all the previous problems. This can create serious problems in isolating faults because no single fault will contain all of the abnormal symptoms that are observed. Fault isolation is the most difficult when the multiple faults result in overlapping symptoms.

Multiple faults also introduce the problem of "fault masking": the presence of one fault may make it impossible to even see symptoms to detect or isolate some other faults. Diagnostic systems that make a single fault assumption do poorly at fault isolation when faced with multiple faults. This is because no single fault will explain all of the observed abnormal symptoms. No single fault signature will be close to the combined fault signature of two faults.

When related faults are considered independent by the FD software, it can be a nightmare to sort through which ones are the root cause of the others and get to the bottom of things. As the building or portfolio gets larger and larger, the single fault assumption gets more and more fatal because teams simply don’t have the time.



The nightmare is exacerbated when FD software vendors have a feature that calculates the avoided costs of each of the individual faults. When each fault is calculated individually without considering related faults, system control sequences, and the building’s seasonal operating schedules, the accuracy is compromised and often results in overestimated savings.

Unfortunately, the nightmare gets even worse if the FD software automatically creates work orders in a computerized maintenance management system (CMMS), resulting in individual tickets for each individual fault and inundating the system with non-actionable tasks.

Accounting for System Effects

The second fatal flaw for FD software is when the algorithms don’t account for system effects. Each piece of building equipment is nestled inside of a system of systems. It's easy to deploy fault detection algorithms on a single piece of equipment (or even entire equipment categories) to identify simple faults. It's much more difficult to model the intended control sequences to accurately identify when the equipment is not aligned with complex control sequences and system interactions.

Airside economizer sequences are a great example of complex sequences where FD software often falls short. What's actually controlling that OA damper? Is it drybulb, enthalpy, outdoor temp? What are the high and low limits that determine when the damper should be at minimum or maximum fresh air? When does it get overridden by humidity controls? What is the minimum outside air fraction supposed to be and is it being reset based on CO_2? And if we look back at our example AHU_03, we’ll see that weakness confirmed: the software failed to detect an issue in the economizer sequence that was exacerbating the High Discharge Air Temperature fault.

When I picture O&M teams using FD tools to move from reactive to proactive, I also picture all the help they'll need in order to translate all the information into action and reportable results. The end user of these tools has 1000s of faults they're attempting to prioritize, diagnose, and turn into action manually. That overwhelm often needs to be overcome by a team of highly trained and specialized engineers who perform those tasks manually. Let's take a look at the work required to investigate faults and get to an action:

Prioritization: The engineer needs to sort through 1000s of faults using filtering and ranking tools (and even exporting to Excel). For the faults that make it through the filter, the engineer must dig into the trend data for each one. Is a leaking valve a priority fault? Not necessarily. If it's January in Boston, HW valve leaking doesn't matter because the heat is needed anyway. This is not sustainable. Prioritization of faults requires accurate cost calculations to be effective.

Diagnosis: The engineer needs to examine trends for multiple faults at the equipment and system level and determine which one is the root cause and which ones are actually symptoms. Then, the likely root cause can be prescribed.

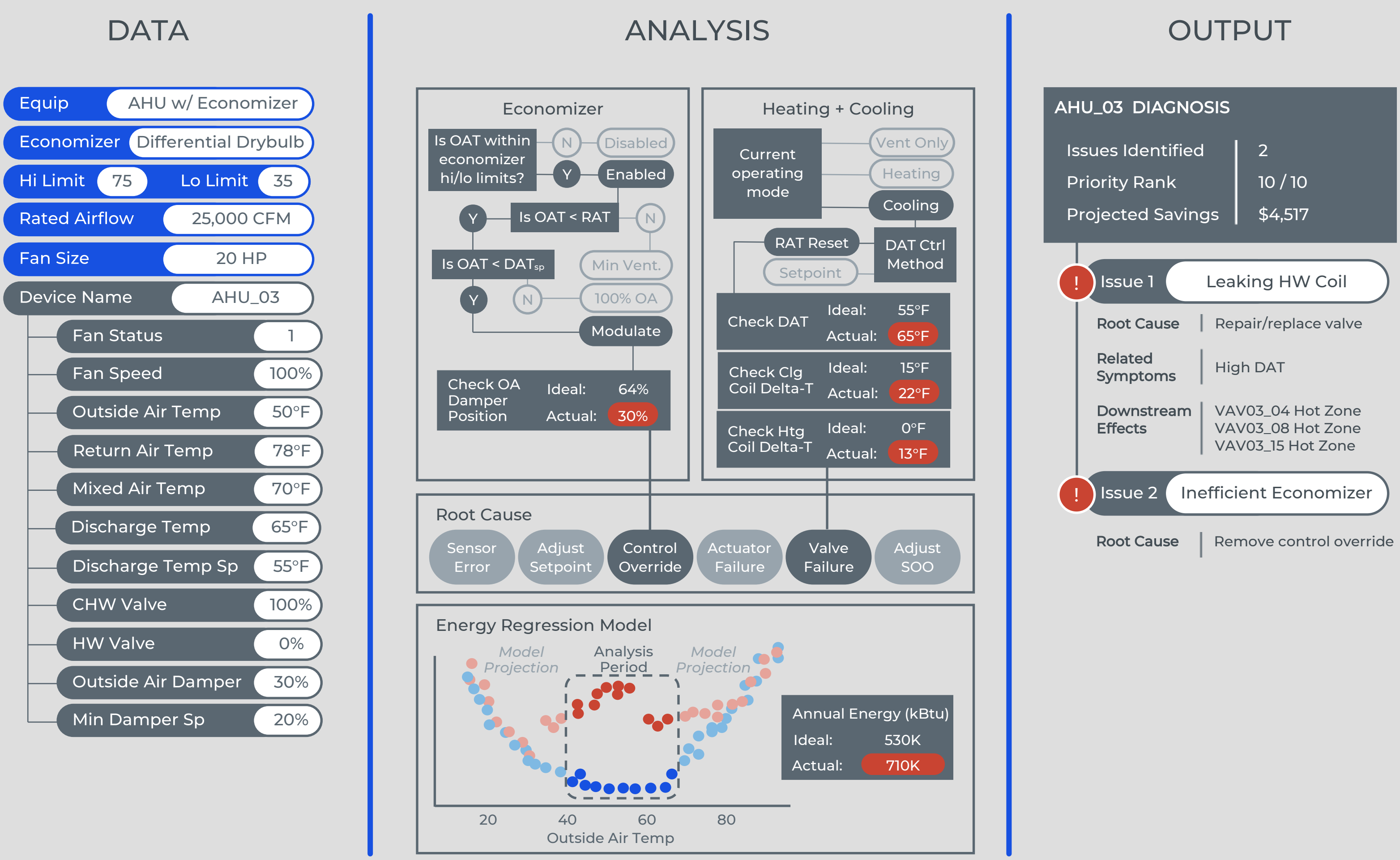
As highlighted in the example above, a single "High Discharge Air Temp" fault could take hours to prioritize and diagnose manually. If the team could automate all of that, they'd be freed up to focus on the important part: taking action. Unfortunately, most of the FDD tools on the market are actually just FD tools. To meet building owners' needs, there's far more to do. Let's proceed.

FAULT DETECTION + DIAGNOSTICS

According to Merriam-Webster, the definition of diagnosis is, “the art or act of identifying a disease from its signs and symptoms”. This definition is pertinent for this paper for two reasons. First, “signs and symptoms” is a really succinct way of summarizing the output of FD platforms. Most “faults” are really just symptoms that might lead to a diagnosis. Second, the use of the word “art” is surprisingly appropriate. As we’ll see soon enough, doing FDD well requires just as much art as science. As much teamwork as automation. As much software design expertise as analytical horsepower.

To illustrate the quantum leap forward provided by the second D, let’s return to our example on AHU_03.

What does Fault Detection & Diagnostics tell us about the AHU issues?



Fault diagnostics means performing an in-depth analysis and pinpointing one or more root causes of problems, to the point where corrective action can be taken. The investigation stage that's required by FD-only tools has been all but automated away.

To dive deeper into this, look at the Economizer Analysis block. The FDD software takes metadata into account that categorizes the economizer as one that is controlled based on the dry bulb temperature differential between return and outside air. Then the algorithm answers a series of questions and determines that the ideal outdoor air damper command is 64%, more than double the actual command, which must be placed in operator override mode (the root cause).

Next, consider the heating and cooling analysis. The FDD software determines the current operating mode, identifies the ideal discharge air temperature, and detects that the actual DAT is too high. It then analyzes historical data on the heating and cooling coils to narrow in on the heating coil as the root cause.

The diagnostic results are summarized in a single diagnostic report for the entire AHU. As expected, there are two issues: Leaking Hot Water Coil and Inefficient Economizing. Each issue has a root cause, related local symptoms, related downstream symptoms, and annual avoidable energy consumption and costs that result from the issue. As you'll see, the diagnostics process is far easier said than done.

Let's unpack the three main steps required fulfill the full potential of the second D:

- Analysis Accuracy
- Automated Root Cause Analysis
- Automated Prioritization

Analysis Accuracy

The Achilles heel of any analytics solutions is the false positive. A building operator will instantly toss your software to the side if you direct them to an issue that turns out to be normal equipment operation, bad data, or a symptom of a larger issue.

The first component of the diagnostic process is to reduce the number of faults on the list, but increase the quality of each one. This requires intelligent algorithms fed with the richest information available. The system needs to understand the attributes of the system and contextualize each fault on the list. For example, if one fault identifies that an air flow or static pressure sensor is reading below its setpoint for an extended period of time, other faults should take that finding into account before they run. If an AHU has a consistent high fan speed, we may not want to identify an opportunity for a static pressure reset to be programmed when the fan is clearly struggling to keep up.

Next, the algorithms should take the design sequence of operations into account. For example, if the system notices an enthalpy setpoint for the economizer, it should infer that the sequence of operations is enthalpy-based. Most end users of FD tools are forced to create special exceptions for quirks like this—essentially a brute force mimicking of the sequence of operations. Again, this sort of one-off customization isn't scalable and the lack of data intelligence must be made up by intelligent human labor.

Finally, in order for this to be scalable, repeatable, and maintainable across a portfolio of diverse systems, the algorithms must dynamically accept the best available information for use in these analyses and checks. For example, when an analysis of the AHU needs to know if the unit is operating it needs to account for all possible methods of determining the "proven on" status: AHU Status, AHU Command, Fan Status, Fan Speed, Discharge Airflow, Discharge Pressure, etc. The analysis can then select from the best available combination of points to produce a consistent diagnostic output across every AHU controller it encounters.

All of these bits of intelligence are required to filter out false positive faults, leaving only the cream of the crop opportunities for the building operator.

Automated Root Cause Analysis

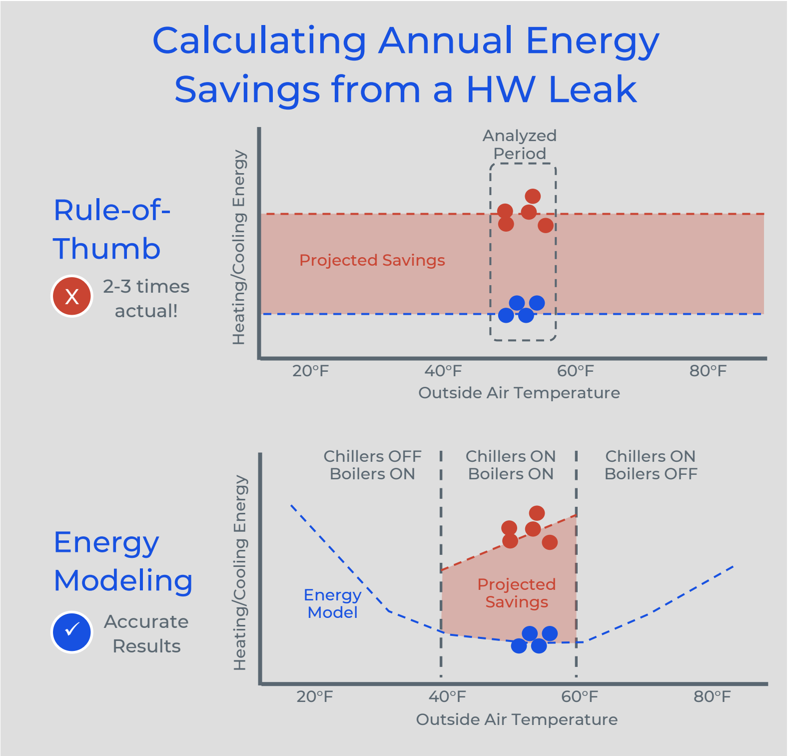
The second component of the diagnostic process is to automatically and dynamically determine one or more root causes at the heart of each fault analysis. All of the data and information are boiled down to a few recommended actions. For AHU_03, these actions were "Repair/Replace Hot Water Valve" and "Remove OA Damper Override".

This is a beautiful thing in and of itself, but the best diagnostics tools I've seen don't stop there. They also provide an explanation to the user on how the algorithms reached their conclusion. Ideally, the user has full confidence in why a particular conclusion was reached. Visibility of all the knowledge that is built into the system is helpful in building that confidence. This also helps build trust before sending a technician to the field to fix the issue.

Automated Prioritization

As previously mentioned, prioritizing faults can be an extremely laborious task. The third requirement for the jump from detection to diagnostics should automate that step by quantifying the impacts of each fault, including costs, energy, comfort, and maintenance impacts.

The cost calculations should avoid rule-of-thumb estimates and leverage accurate energy modeling techniques whenever possible. Static variables (e.g. fan HP, rated flow) and dynamic variables (e.g. live point information such as fan speed, flow) should be used to calculate avoidable kBTU's, kWh, cooling ton hours, and other energy variables and convert those metrics into avoidable costs based on both blended and complex rate schedules. Monthly built-in automated measurement & verification (M&V) algorithms should annualize the savings calculations to make each fault even more actionable by gauging significance relative to other O&M costs and allowing return-on-investment comparisons.



As a result of these three components of diagnostics algorithms, the list of diagnosed faults is much shorter and more actionable than FD-only tools are capable of. Root cause and investigation and verification are done by the tool, meaning your team can spend more time on resolution. For all the nerdy talk about algorithms, that's the true value of diagnostics: freeing people up to focus on the most valuable part of the FDD process, which is taking action and improving performance. Since the heavy lifting is done by the tool, the O&M team's focus can now shift towards getting real stuff done.

Teams, which include service contractors and technicians, can unite around common KPIs (e.g. number of fixed faults or energy use avoided) and custom reporting or dashboards so everyone can work toward a common goal. The teams can also interact around the implementation of each diagnosed fault. Or, the FDD tool can integrate directly with the CMMS to drive work orders and ultimately action. Since the faults have been fully vetted by the diagnostics algorithms, this integration can be deployed with confidence.

FAULT DETECTION +

DIAGNOSTICS CHECKLIST

Given the value of the second D, why do most FDD platforms not go all the way? Because it's much more difficult. Diagnostics, as we've seen, require a software design that is very intentionally built for scalability and accuracy across hundreds of thousands of systems. It also requires rich contextual information on how each system was designed, which takes effort to collect equipment schedules, control sequences, and design drawings needed to actually model how the systems should be operating.

It's time for a challenge for our industry: just because something is hard, doesn't mean we shouldn't do it. Building owners need FDD tools that go the full distance to the second D.

A myth has been propagated in the community of early adopters of buildings analytics technology. The myth says that because analytics tools require action by a "human in the loop", the people and processes involved are the only thing that matters. As we've covered in this paper, how much that human can get done is 100% determined by the effectiveness of the analytics platform employed – is it only FD, or truly FD&D? There's actually a limit to how much you can accomplish with people and processes if the analytics tool hasn't been carefully selected. As we've seen, FDD can be accomplished with toolkits that are most commonly used for FD-only, but they require too much manual effort, overwhelm the O&M team, and delay the path to action and real results.

So if you're ready to transition from reactive maintenance to proactive maintenance make sure you're equipped with the right tools. No FDD platform is complete until it checks all of the boxes in the FDD checklist:

FAULT DETECTION +

DIAGNOSTICS CHECKLIST

✓ Analysis Accuracy

- Demonstrated ability to suppress false positive faults based on other fault conditions
- Rules are aware of each other
- Rules are implemented into a hierarchy

✓ Automated Root Cause Analysis

- Possible causes are dynamically generated by extensive logic based on the best available information. This means if there are more sensors or less, diagnostics can still produce valuable results.
- Root causes include an explanation to the user on how the algorithms reached their conclusion

✓ Automated Quantification of Impacts

- Impacts include cost, energy, maintenance, and IEQ impact
- Automated Measurement and Verification (M&V) is based on standard engineering and can be demonstrated transparently