Fault Identification Using System-Level Insights and Multi-Layered Classification

Ryan Aalund[1,2], Meenusree Rajapandian[2], Vincent P. Paglioni[1]

[1]Risk, Reliability, & Resiliency Characterization Lab, Department of Systems Engineering, Colorado State University, Fort Collins CO 80523.

ryan.aalund@colostate.edu,

vincent.paglioni@colostate.edu

[2]Samsara Inc, San Francisco CA 94107.

meenusreerajapandian@gmail.com,

ABSTRACT

This paper presents a novel framework for IoT device fault detection, combining meta-algorithmic decision logic with a neural network classifier for efficient failure analysis. Utilizing system-level data, the methodology employs a multi-layered architecture to classify devices as either failed or non-failed and identify the root cause, whether hardware, software, or firmware. The first layer implements a meta-classifier that integrates multiple lightweight algorithms weighted by application-specific criteria such as accuracy or recall. This ensemble approach enhances fault detection by utilizing high-level system metrics. The second layer introduces a neural network trained on subsystem-specific features, such as power metrics and diagnostics, to infer the most probable root cause. This structure enhances the accuracy of failure prediction while improving the interpretability of device failures and their potential root causes. Demonstrated on real-world telematics devices collecting GPS data, the framework addresses the need for scalable diagnostic methods in high-volume environments. By minimizing unnecessary returns and streamlining workflows, this approach delivers practical value in field operations. The modular two-tiered architecture allows for adaptability to various device types and fault modes. Future work will explore model generalization across different deployments and enhance root cause analysis through structured, data-driven methods to improve operational reliability of the framework.

1. INTRODUCTION

As connected systems become more integral to critical infrastructure, logistics, and operational automation, the ability to reliably detect and diagnose device faults at scale has emerged as a core requirement in system reliability engineering (Leite, Andrade, Rativa, & Marciel, 2025). Modern embedded and internet-of-things (IoT) systems, often deployed in high-volume, distributed environments, are expected to operate autonomously for extended periods of time. Despite typically low failure rates, the operational and economic cost of missed failures, or the unnecessary removal of healthy devices, can be substantial (Stergiopoulos, Kotzanikolaou, Theocharidou, Lykou, & Gritzalis, 2016). This challenge is exacerbated by limited visibility into subsystem-level degradation and the complexity of interdependent failure mechanisms (Sinha & Lee, 2024).

Traditional approaches to fault detection have primarily relied on either single-metric, rule-based heuristics or monolithic classifiers trained on narrow feature sets (Gertler, 2017). These techniques often lack both the interpretability and modularity required for deployment in evolving system architectures. They frequently assume static behavior and fail to consider the nuanced trade-offs between different performance metrics, such as accuracy, precision, and recall, which are crucial in real-world decision-making contexts. That is, typical fault detection approaches lack applicability in the era of extreme complexity and interdependency.

This paper presents a structured fault detection and diagnosis (FDD) framework that integrates meta-algorithmic decision logic with a neural network-based classifier to deliver robust

and interpretable device health assessments. The methodology employs a multi-layered classification architecture rooted in systems thinking, where the first layer determines the device failure status using system-level metrics, and a second layer identifies the subsystem from which the failure originates. This modular approach supports both high-confidence failure detection and granular root cause insights while remaining adaptable to data constraints and deployment variability.

The first classification layer employs a custom-weighted meta-classifier that integrates multiple lightweight algorithms. These algorithms are weighted not solely by accuracy, but by application-aligned metrics such as precision or recall, depending on the business or safety context. This enables the system to prioritize operational objectives, such as minimizing false negatives in safety-critical applications or false positives in return-sensitive logistics.

The second layer introduces a neural classifier trained on features extracted from subsystems, including battery voltage, system-on-chip (SoC) diagnostics, and LTE module behavior. In the current implementation, this classifier uses time series metrics to capture interactions between subsystems and support root cause inference. The architecture is designed to be extendable, with future iterations built on recurrent or convolutional neural networks (RNN/CNN) to process raw time-series data directly and utilize the complete available dataset, including partially labeled data, thereby enhancing the system's ability to detect intermittent and dynamic failures (Jung, Han, & Choi, 2021).

This paper develops the approach and demonstrates its utility on telematics devices used in vehicular environments. However, the methodology is broadly applicable to any system requiring fault detection and identification, particularly those operating at scale with complex subsystem interactions and limited access to labeled failure data. By structuring failure detection into two stages and optimizing each layer for interpretability, adaptability, and diagnostic value, this framework enables a scalable path toward more autonomous and data-driven reliability management.

The remainder of this paper is organized as follows: Section 2 reviews related work in fault detection, ensemble classification, and neural diagnostic systems. Section 3 formulates the fault detection problem and outlines the system constraints. Section 4 describes the layered methodology and feature engineering process. Section 5 presents experimental results based on a real-world dataset. Section 6 discusses model insights, unexpected findings, and limitations. The paper concludes with Section 7.

2. RELATED WORKS

The application of Prognostics and Health Management (PHM) and FDD techniques to IoT systems has become an increasingly critical focus as connected devices proliferate across industrial, commercial, and infrastructure environments. IoT systems present unique challenges for PHM due to their distributed architecture, constrained resources, variable data quality, and limited access to physical inspection. These factors demand FDD strategies that are robust, lightweight, and capable of operating under partial observability and intermittent connectivity.

Several reviews have emphasized the need for tailored FDD approaches in IoT-enabled environments. Leite et al. (2025) and Abid, Khan, and Iqbal (2021) outline the growing complexity in deploying traditional FDD methods in modern cyber-physical systems, highlighting that standard diagnostic models often fail to generalize across the heterogeneous and dynamic conditions seen in IoT deployments. Chi, Dong, Wang, Yu, and Leung (2022) further note that many IoT systems lack sufficient labeling or historical fault data, making knowledge-based and unsupervised FDD more attractive despite their limitations in interpretability and domain portability.

The scale and variability of IoT deployments have led researchers to explore distributed and hierarchical fault identification architectures. Marino, Wisultschew, Otero, Lanza-Gutierrez, Portilla, and de la Torre (2021) introduced a distributed machine-learning system that adapts FDD quality based on the context and resource constraints of edge devices. Similarly, Aldaajeh, Harous, and Alrabaee (2021) proposed FDD design tactics that optimize embedded system efficiency, striking a balance between computational load and diagnostic performance.

AI-enabled fault diagnosis has become a dominant trend in IoT FDD, as explored in surveys by Nguyen, Medjaher, and Tran (2023) and Lo, Flaus, and Adront (2019). However, deployment remains challenging due to data sparsity and system variability. Sinha and Lee (2024) argue that despite significant advances in lab-scale model performance, field deployment in IoT systems is hindered by unresolved issues in lifecycle management, domain shift, and model explainability. This is reinforced by real-world studies such as that of Dzaferagic, Marchetti, and Macaluso (2022), who address sensor dropout in industrial IoT (IIoT) through data imputation using generative adversarial networks, enhancing fault classification resilience under missing data conditions.

Edge-focused FDD strategies have also gained traction, particularly in situations where cloud latency or data bandwidth make centralized PHM impractical. Lu, Lu, An, Wang, and He (2023) and Hadi, Hady, Hasan, Al-Jodah, and Humaidi (2023) explore edge-deployable diagnostic pipelines and AutoML techniques to reduce model

development burden while enabling responsive diagnostics. These strategies facilitate fault detection to be colocated with the device, improving latency and reliability.

Other works have explored domain-specific FDD implementations in IoT systems such as solar energy (Balakrishnan, Raja, Sudhakar, & Janani, 2023), smart grids (Al Mhdawi & Al-Raweshidy, 2020), marine equipment (Orhan & Celik, 2024), and industrial robotics (Raouf, Kumar, Lee, & Kim, 2023). These studies consistently demonstrate that IoT system characteristics include limited telemetry, physical inaccessibility, and variability in fault expression. These characteristics challenge traditional FDD approaches; therefore, applying FDD to IoT systems requires rethinking how FDD models are trained, validated, and deployed. Lavanya, Prasanth, Jayachitra, and Shenbagarajan (2021) exemplify this by developing a tuned classification method for FDD in wireless sensor network (WSN)-based IoT systems that copes with signal heterogeneity and fault ambiguity.

Recent literature also emphasizes that classification accuracy alone is insufficient as a benchmark for PHM in IoT contexts. Fault models must be tuned to prioritize specific error tradeoffs depending on operational cost structures, for instance, minimizing false positives in logistics (Seabra, Costa, & Lucena, 2016) or false negatives in safety-critical controls (Kim & Katipamula, 2017). Researchers have begun adopting cost-sensitive or meta-learning models to manage this, though generalized frameworks are still lacking.

Overall, the state of FDD for IoT systems reflects a field in active transition. Early rule-based approaches and expert-defined diagnostics are giving way to learning-based and context-aware systems; however, new constraints imposed by the IoT environment, such as imbalanced data, missing signals, and energy constraints, necessitate methodologies that evolve beyond traditional assumptions. The existing body of work has laid the basis for foundational approaches, but few offer unified, modular architectures capable of adapting across deployments, fault types, and operational priorities.

3. PROBLEM FORMULATION

Fault detection in embedded and connected systems presents a challenging trade-off between detection sensitivity and operational precision (Aldaajeh, Harous, & Alrabaee, 2021). While failures may occur infrequently, the consequences of undetected faults or unnecessary field actions can be significant (Smith, 2021). This is particularly true in large-scale deployments where even a small failure rate translates into thousands of potentially impacted devices, and where operational decisions must be made with limited diagnostic access to the physical system (Lwakatare, Raj, Crnkovic, Bosch, & Olsson, 2020).

The classification task addressed in this work involves two critical objectives:

1. Determining whether a device has failed, based on system-level performance indicators.
2. Identifying the most probable root cause domain (e.g., hardware or software), based on observable subsystem behaviors.

The motivating dataset for this study is drawn from proprietary telematics devices deployed at scale. These devices collect trip data via onboard GPS modules and transmit information over LTE networks to cloud infrastructure (Ghaffarpasand, Burke, Osei, Ursell, Chapman, & Pope, 2022). Although the annual observed failure rate is low (less than 1% over three years), the volume of deployed units yields sufficient labeled examples for machine learning-based fault detection. However, the general problem formulation applies to a wide range of IoT, embedded, and cyber-physical systems where devices operate autonomously and faults emerge through indirect symptoms (Smart, Grimm, & Hartzog, 2021).

To address this classification problem, we adopt a multi-layered approach. The first layer is responsible for high-confidence failure detection, utilizing features that reflect overall system functionality, such as signal acquisition uptime or communication health. The second layer focuses on failure classification, drawing on subsystem-level data (e.g., battery metrics, processor diagnostics, communication module behavior) to infer from which subsystem the failure originates. In the current implementation, a limited number of subsystems are included: power, physical, firmware, GPS, and LTE connectivity. However, future work will aim to include more subsystems.

This layered structure is motivated by practical considerations. It is more critical to correctly identify that a device has failed than to diagnose the underlying cause immediately (Okes, 2019). Misclassifying a failed device as functional could allow for continued degradation and customer impact. Conversely, incorrectly labeling a functional unit as failed incurs unnecessary return costs and logistics overhead (Wilson & Goffnett, 2022). By decoupling detection from diagnosis, the model is able to prioritize early, conservative identification of failure while preserving interpretability and adaptability in downstream classification.

Additionally, this structure facilitates scalability: as richer telemetry or more granular tagging becomes available, the diagnostic classifier can be retrained or replaced without altering the upstream detection layer. This modularity enables the deployment of the framework across various system types, making it suitable for any operational environment where low-visibility failures, high-volume field data, and domain-specific performance constraints define the fault detection challenge.

4. METHODOLOGY

4.1. Framework Architecture and Design Philosophy

The proposed two-layer classification framework is designed to strike a balance between fault detection sensitivity and diagnostic resolution. The layered design is rooted in systems thinking, separating fault detection (Layer 1) from fault diagnosis (Layer 2) to address each task with appropriate methods and data representations.

Layer 1 performs binary classification to determine if a device has failed. It utilizes a lightweight, interpretable meta-algorithmic decision engine that relies on system-level indicators and derived metrics (Simske, 2013). Layer 2 focuses on identifying the most likely root cause domain — power, connectivity, GPS, firmware, or hardware-related issues — through a neural classifier trained on subsystem-specific features.

This separation enhances scalability and interpretability. If a new diagnostic signal becomes available or if the classification priority changes (e.g., from minimizing false positives to maximizing detection coverage), only the corresponding layer needs to be adjusted. This separation offers multiple advantages, as it decouples critical detection sensitivity from the less reliable but still valuable diagnostic inference; it improves modularity, allowing updates to one layer without affecting the other; and it supports interpretability, enabling Layer 1 to be tuned to minimize high-cost misclassifications (e.g., false negatives), while Layer 2 can evolve as richer features or labels become available. The framework is designed to adapt across deployments with different system architectures, failure modes, datasets, and/or operational goals.

4.2. Data Sources and Preprocessing

Data is collected from deployed telematics devices, comprising both system-level metrics, subsystem-specific telemetry and user interactions:

- **System-level metrics:** LTE uptime (%) GPS successful fixes (%), system reboots;
- **Subsystem metrics:** Battery voltage, SoC temperature, connectivity statistics; *and*
- **User metrics:** Total time in field, time since last activity, and other user interactions like button presses, user functional modes like frequency of check-ins.

To simplify the classification process and address constraints in real-time systems, telemetry streams are transformed into *snapshot metrics* (e.g., averages, uptime ratios, etc.). This approach avoids the complexity of processing time-series data and enhances the interpretability of any model built upon it, although this tradeoff is discussed in Section 6. Converting time-series telemetry data into snapshot metrics reduces the dimensionality of the problem space. In this work, the following transformations are applied for each device:

- **Statistical aggregation:** e.g., mean uptime over a rolling window;
- **Derived features:** e.g., battery voltage × SoC temp, LTE uptime ÷ GPS uptime;
- **Normalization:** Features are z-scaled or normalized between [0, 1] as appropriate; *and*
- **Outlier filtering:** Extreme values are removed based on percentile or domain heuristics

Dataset Construction: Devices are labeled as failed, along with a possible root cause based on customer complaints, return analysis, and fault investigation. The failure information is used in layer one, and the root cause is in Layer 2. While there are no clearly labeled devices as functional, any device that has not failed is considered functional. While this may introduce error in our dataset, where devices labeled as functional have failed in the field, this is expected to be very small, given the already low rate of failure. The dataset was randomly partitioned into training (80%) and testing (20%) sets.

System-level device performance metrics, subsystem snapshot metrics, and user interaction metrics were used in the model.

- LTE uptime (%) - percent of time connected to the network
- GPS Fix – percent of GPS fixes that resulted in a valid location
- Reboots – total number of power reboots the system underwent, both initiated by a user and by the firmware to reboot the system
- Total GPS check-ins attempted by the device
- Battery voltage levels
- Total time the device was active in the field
- Maximum number of button press reboots initiated by the users in a day
- Functional mode of the device – check-in frequency

The model was initially trained using both the original and normalized features. Eventually, features were selected based on their accuracy in prediction and their significance in determining the classification of labels. A total of 2916 failed devices were used in the dataset, and 3000 functional devices were randomly selected from the larger population of 220,000 devices to train the model. Note that the features on functional devices exhibited very low variance (see results in Figures 1 and 2), and any random selection of 3000 devices was representative of the entire field population. While the model is independent of the sample size of each class (if there are enough to represent the field population), this sampling was done to minimize the effect of failed devices that are

mislabeled as functional. This was done instead of removing outliers in the features, as a number of these outliers represented actual functional or field failures.

4.3. Layer 1 – Device Failure Detection

Layer 1 is optimized to identify whether a device is exhibiting failure symptoms by utilizing system-wide indicators. It uses a lightweight, interpretable meta-algorithmic decision engine (Simske, 2013) that identifies the critical point separating the two classes in each feature. Each simple classifier finds a critical point for each feature that divides the feature set into values closer to the mean value of its corresponding class label. These simple classifiers are weighted based on the chosen classification metric, giving more importance to critical point thresholds that contribute to a higher classification metric. This results in the label for the corresponding device being marked as either 'failed' or 'functional'. These indicators are both accessible and interpretable in large-scale deployments, making the solution deployable with minimal computation.

Classification Method: A meta-algorithmic decision model is employed, which combines multiple lightweight classifiers or decision rules, each evaluated based on its performance with respect to selected classification metrics (see Section 4.5). The output is a binary classification:

- Class A: Device has failed
- Class B: Device is functional

4.4. Layer 2 – Root Cause Domain Classification

For devices classified as failed, Layer 2 attempts to infer the subsystem at fault. The classifier used is a shallow neural network, chosen for its ability to capture non-linear interactions while remaining computationally light.

The feature set includes the same features as Layer 1, but instead of a snapshot, it uses the last 10 days' time series for each corresponding feature. The result of Layer 2 is to find the subsystem at fault, given that Layer 1 resulted in a failure classification. The result of this layer is one of the following labels: power, physical, firmware, GPS, and LTE connectivity.

4.5. Meta-Algorithmic Classifier Weighting

A novel contribution of this framework is the application-driven weighting of classifiers in Layer 1. Instead of optimizing solely on accuracy, the framework allows weights to be tuned based on precision, recall, or F1-score, depending on the operational consequences of each type of error. In Table 1, examples of organizational impact are shown, illustrating how each metric can be applied.

The meta-classifier calculates classifier influence using a weighting vector derived from historical performance across these metrics. For instance, in a high-precision deployment, classifiers producing fewer false positives are weighted more heavily, even if their overall accuracy is lower. This adaptability enables the model to be re-optimized dynamically in response to deployment changes, without requiring retraining of the underlying classifiers.

Table 1. Rationale for Metric-Based Weighting

Metric	Prioritized When...	Example Contexts
Accuracy	All error types are equally costly	General monitoring in non-critical systems
Precision	False positives must be avoided (e.g., avoid returning healthy devices)	Logistics/cost-sensitive operations
Recall	False negatives are unacceptable (e.g., missing a failed safety-critical device)	Vehicle control, healthcare, or safety systems
F1-Score	Balanced treatment of both precision and recall when tradeoffs are unclear	Early-stage models with limited ground truth

The Benefits of this approach include operational alignment, where model decisions reflect real-world impact. Modularity is facilitated by allowing weights to be updated independently of feature sets or classifier design. Interpretability is built by tracing each classifier's contribution to a performance rationale.

4.6. Evaluation Strategy

Performance metrics for both layers include accuracy, precision, recall, F1-score, and a confusion matrix (a multiclass version is used for Layer 2). Additional robustness tests involve T-tests to assess distributional drift between training and testing data.

5. RESULTS

5.1. Layer 1: Fault Detection Performance

All the features listed in Section 4.2, along with normalized and derived metrics, were initially used in the model. However, only four features were significant in classifying the devices: (1) time of the last activity, (2) battery levels, (3) total system reboots, and (4) the firmware version the device was running. Figures 1 and 2 show the results on classification metrics across multiple sample sets of 3,000 functional devices from a population of 220,000 devices. The results discussed in this layer are based on 20 such samples. The critical points for each classifier across these samples had a difference of < 10% (weighted for each feature), confirming that the samples were in fact representative of the field population.

As shown in Figure 1, the accuracy of the test data remained high throughout. However, when weighted on accuracy, the precision score is low. This may be due to failed devices being incorrectly marked as functional, likely because no failure signal was received from the field for those devices. When weighted by precision, some improvements in precision and F1 scores are observed, with minimal changes in accuracy scores. This demonstrates how the model can be tuned to achieve business-specific objectives. Note that the recall is very high when weighted against accuracy, but decreases when using precision, as shown in Figure 2. This is because, as the focus is on reducing false positives, false negatives also increase, leading to a reduced recall value. This is likely due to the non-linear separability of classes on these features. While the focus here is on creating a simple and computationally inexpensive classifier, changing the classification metrics used to weight the simple classifiers has been shown to be sufficient for this purpose, and no exploration of non-linear classification was conducted.

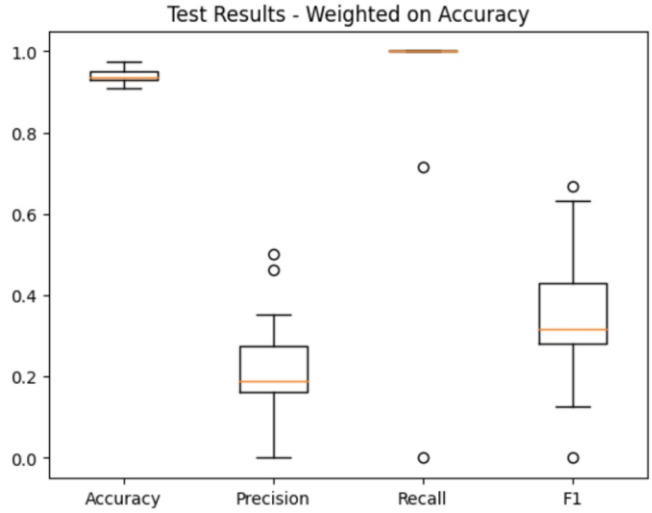

Figure 1. Classifier results weighted on accuracy

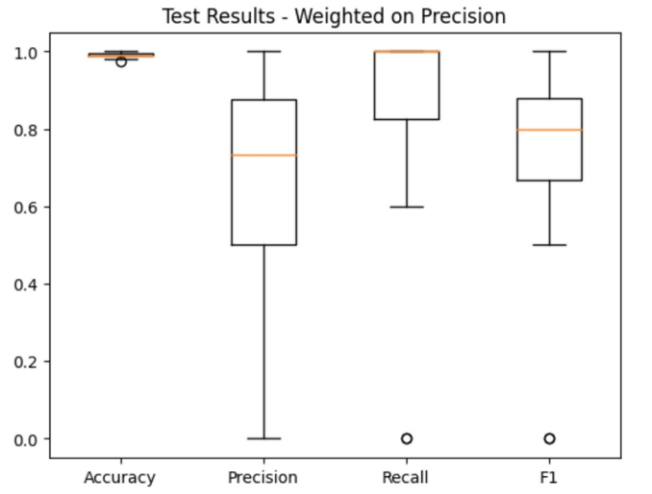

Figure 2. Classifier results weighted on precision

5.2. Layer 2: Root Cause Classification Performance

A simple neural network was used for this layer, utilizing the last 10 days of device data for each feature instead of snapshot metrics. The multivariate time series data was converted to a univariate stream of information and provided as input to the neural network. Only devices that were known failures were used to train this 4-layer model. The reduced number of layers was picked to avoid overfitting the data. An additional 108 devices were removed from this model due to known rare failure modes. These removals would have unnecessarily increased the model's complexity with additional low-value classes. Although the data set is large (2,809), it is somewhat limited due to the wide variety of failures observed in the field. Each class in this model has a distinct signature of failure. For example, “Battery” is a general classification for all battery-related problems; however, loss of battery capacity is only one of the many ways that could result in this label. It could also be due to repeated FW loops getting stuck or other subsystem interactions. There is high confidence in the labels, since a majority of these devices were physically screened to identify the root cause. Despite this, the classification metrics are likely low due to the limited data used to train the model and the limited time series information incorporated into the neural network. Devices experience cumulative degradation of their hardware and software, which may result in failure. Repeated exposure to stress is a significant cause of the majority of device failures in the field. The 10-day data provides a limited view of all the stressors to which the device was exposed.

Table 2. Predicted versus actual performance (Confusion Matrix) of Layer 2

	Predicted					
ACTUAL	Battery	Firmware	GPS	System Power	Water Ingress	Total
Battery	270	7	0	189	5	471
Firmware	0	1019	0	141	0	1160
GPS	17	0	0	89	1	107
System	364	0	0	592	5	961
Water Ingress	66	0	0	48	5	119
Total	717	1026	0	1059	16	

The mean accuracy for this model was 66.7%. Although low, the classified labels themselves provide several observations. Precision and recall were highest for firmware issues, whereas GPS classification performed the worst. This could be because, while the most recent firmware information is sufficient for detecting firmware issues, GPS issues are likely intermittent, and a 10-day historical view is likely insufficient to detect those issues. Water ingress issues are challenging to

detect using data independent of model type, as water can impact different subsystems based on installation location and user function, resulting in both low precision and recall, and potentially leading to high misclassification rates.

6. DISCUSSION

The final model metrics for the tuned Layer 1 were 94% accuracy, 67% precision, and 84% recall across the test sets. This model resulted in high values on all measure classification metrics. However, the model is designed in a way that, in the event of unbalanced values, tuning is simple and computationally inexpensive. It also renders itself well in cases where finding the result of each of these cases and understanding the data distribution better is required.

Table 3. Layer Two Performance

Feature	Precision	Recall
Battery	37.6	57.3
Firmware	99.2	87.8
GPS	0	0
System Power	55.5	61.2
Water Ingress	31.3	41.6

Having control over the tuning of these weights is beneficial because it ensures that the outcome of the failure classification is helpful for both the application and the business use case. This approach also eliminates the limitation of needing to identify the correct root cause before realizing value. Future evaluations will benchmark stage-level outputs and the end-to-end system to quantify both incremental and overall value.

Layer 2 achieved only moderate performance, with 66% accuracy. This is partly because identifying particular subsystems that failed is complex, requiring a historical view of time series data from different sensors on the device. However, only 10 days of data were used in this work. Furthermore, using RNNs or CNNs that possess temporal memory and local pattern detection capabilities is expected to significantly improve the performance of this layer. In addition to architectural enhancements, Layer 2 will be rigorously benchmarked against a linear classifier and a prototype-based classifier on the same feature set to substantiate its incremental value and delineate operating regimes where it wins or loses.

These results suggest that although the subsystem metrics used (battery voltage, SoC temperature, and LTE performance) are directionally useful, they lack the necessary granularity and separability to reliably assign failure domains in all cases.

Additionally, models in both Layers 1 and 2 assume that devices are either functional or failed. While there is a reliable signal for failed devices in the field, such as customer complaints, warranty exchanges, or complete inactivity of devices, there are often unknown devices that are partially functional, with some subsystems failing. While these are assumed to be functional for these models in both layers, considering data as partially labelled will significantly improve the models, especially for Layer 2, which is expected to detect failed subsystems.

Although the model showed no signs of overfitting, expanding the dataset and refining the feature-label alignment would likely improve the overall model's generalization.

In summary, the experiment confirms that a multi-layered classification structure, with system-level detection followed by subsystem-guided diagnosis, offers a practical and interpretable approach to fault identification. While Layer 1 is production-ready in its current form, Layer 2 reveals several opportunities for refinement in labeling, feature development, and evaluation strategy. These insights lay the groundwork for the next phase of system improvement and model generalization. Planned baselines for Layer 1 include head-to-head comparisons with logistic regression and XGBoost to contextualize performance. Results will be reported at the layer level and for the whole pipeline, with standardized win/loss analyses to demonstrate incremental gains from each stage and the aggregate benefit of the overall system.

7. CONCLUSION

This paper introduces a fault identification framework that employs a multi-layered classification approach, grounded in system-level metrics and subsystem insights. The design decouples fault detection from diagnosis, allowing each to be optimized independently and aligned with operational priorities. The first classification layer focused on identifying whether a device had failed, demonstrating high reliability with 94% accuracy and substantial precision–recall balance. This layer successfully leveraged lightweight features and derived metrics to maximize separation between functional and non-functional devices while minimizing false positives.

A core innovation in this framework is the use of application-driven classifier weighting, where performance metrics such as accuracy, precision, or recall are explicitly used to tune classification behavior. This method supports operational decision-making by aligning algorithmic decisions with the cost structure and risk tolerance of the field environment.

While Layer 1 is suitable for deployment in systems requiring high-confidence failure identification, Layer 2 offers directional guidance but would benefit from additional refinement. Future development will focus on several areas:

- Incorporating more granular failure categories, including a distinct firmware class

- Enhancing feature sets to capture diagnostic signals and temporal context better
- Addressing class imbalance through dataset expansion or rebalancing techniques
- Improving label fidelity through deeper investigation of failure progression and symptoms

Although the experimental results were derived from telematics devices, the methodology is broadly applicable to other connected systems, including industrial, infrastructure, and embedded IoT environments. The framework's modularity and interpretability make it suitable for scaling across platforms and evolving with system complexity. Ultimately, this work lays the foundation for data-driven, system-aware fault identification that strikes a balance between sensitivity, scalability, and actionable insights.

REFERENCES

Abid, A., Khan, M. T., & Iqbal, J. (2021). A review on fault detection and diagnosis techniques: basics and beyond. *Artificial Intelligence Review*, *54*(5), 3639-3664. https://doi.org/10.1007/s10462-020-09934-2

Al Mhdawi, A. K., & Al-Raweshidy, H. S. (2020). A smart optimization of fault diagnosis in electrical grid using distributed software-defined IoT system. *IEEE Systems Journal, 14*(2), 2780–2790. https://doi.org/10.1109/JSYST.2019.2921867

Aldaajeh, S. H., Harous, S., & Alrabaee, S. (2021). Fault-detection tactics for optimized embedded systems efficiency. *IEEE Access, 9*, 91328–91340. https://doi.org/10.1109/ACCESS.2021.3091617

Balakrishnan, D., Raja, J., Sudhakar, K., & Janani, K. (2023). An IoT-based system for fault detection and diagnosis in solar PV panels. In *E3S web of conferences* (Vol. 387, p. 05009). EDP Sciences.

Cao, X., Yang, R., Guo, C., & Wu, X. (2025). An end-to-end framework for fault detection and diagnosis electric rudder servo system-based under imbalanced data condition. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, https://doi.org/10.1177/09544100251342256

Chi, Y., Dong, Y., Wang, Z. J., Yu, F. R., & Leung, V. C. (2022). Knowledge-based fault diagnosis in industrial internet of things: a survey. *IEEE Internet of Things Journal*, *9*(15), 12886-12900.

Dzaferagic, M., Marchetti, N., & Macaluso, I. (2022). Fault detection and classification in Industrial IoT in case of missing sensor data. *IEEE Internet of Things Journal, 9*(11), 8892–8900. https://doi.org/10.1109/JIOT.2021.3116785

Gan, C. L. (2020). Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things: John Wiley & Sons Ltd.(2018). pp. 731, ISBN: 9781119515326 (Print), 9781119515326 (Online). *Life Cycle Reliability and Safety Engineering*, *9*(2), 225-226.

Gertler, J. (2017). *Fault detection and diagnosis in engineering systems*. CRC Press. https://doi.org/10.1201/9780203756126

Ghaffarpasand, O., Burke, M., Osei, L. K., Ursell, H., Chapman, S., & Pope, F. D. (2022). Vehicle Telematics for Safer, Cleaner and More Sustainable Urban Transport: A Review. *Sustainability*, *14*(24), 16386. https://doi.org/10.3390/su142416386

Hadi, R. H., Hady, H. N., Hasan, A. M., Al-Jodah, A., & Humaidi, A. J. (2023). Improved Fault Classification for Predictive Maintenance in Industrial IoT Based on AutoML: A Case Study of Ball-Bearing Faults. *Processes*, *11*(5), 1507. https://doi.org/10.3390/pr11051507

Jung, Y. J., Han, S. H., & Choi, H. J. (2021). Explaining CNN and RNN using selective layer-wise relevance propagation. *IEEE Access*, *9*, 18670-18681.

Kim, W., & Katipamula, S. (2017). A review of fault detection and diagnostics methods for building systems. *Science and Technology for the Built Environment*, *24*(1), 3–21. https://doi.org/10.1080/23744731.2017.1318008

Lavanya, S., Prasanth, A., Jayachitra, S., & Shenbagarajan, A. (2021). A tuned classification approach for efficient heterogeneous fault diagnosis in IoT-enabled WSN applications. *Measurement, 183*, 109771. https://doi.org/10.1016/j.measurement.2021.109771

Leite, D., Andrade, E., Rativa, D., & Maciel, A. M. A. (2025). Fault Detection and Diagnosis in Industry 4.0: A Review on Challenges and Opportunities. *Sensors*, *25*(1), 60. https://doi.org/10.3390/s25010060

Lo, N. G., Flaus, J. M., & Adrot, O. (2019, July). Review of machine learning approaches in fault diagnosis applied to IoT systems. In *2019 International Conference on Control, Automation and Diagnosis (ICCAD)* (pp. 1-6). IEEE.

Lu, S., Lu, J., An, K., Wang, X., & He, Q. (2023). Edge computing on IoT for machine signal processing and fault diagnosis: A review. *IEEE Internet of Things Journal*, *10*(13), 11093-11116.

Lwakatare, L. E., Raj, A., Crnkovic, I., Bosch, J., & Olsson, H. H. (2020). Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. *Information and software technology*, *127*, 106368. https://doi.org/10.1016/j.infsof.2020.106368

Marino, R., Wisultschew, C., Otero, A., Lanza-Gutierrez, J. M., Portilla, J., & de la Torre, E. (2021). A machine-learning-based distributed system for fault diagnosis with scalable detection quality in industrial IoT. *IEEE Internet of Things Journal, 8*(6), 4339–4352. https://doi.org/10.1109/JIOT.2020.3026211

Nguyen, K. T. P., Medjaher, K., & Tran, D. T. (2023). A review of artificial intelligence methods for engineering

prognostics and health management with implementation guidelines. *Artificial Intelligence Review, 56*, 3659–3709. https://doi.org/10.1007/s10462-022-10260-y

Okes, D. (2019). *Root cause analysis: The core of problem solving and corrective action*. Quality Press.

Orhan, M., & Celik, M. (2024). A literature review and future research agenda on fault detection and diagnosis studies in marine machinery systems. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, *238*(1), 3-21.

Raouf, I., Kumar, P., Lee, H., & Kim, H. S. (2023). Transfer Learning-Based Intelligent Fault Detection Approach for the Industrial Robotic System. *Mathematics*, *11*(4), 945. https://doi.org/10.3390/math11040945

Sangeetha, S. B., Mani, P., Maheshwari, V., Jayagopal, P., Sandeep Kumar, M., & Allayear, S. M. (2022). Design and Analysis of Multilayered Neural Network-Based Intrusion Detection System in the Internet of Things Network. *Computational Intelligence and Neuroscience*, *2022*(1), 9423395. https://doi.org/10.1155/2022/9423395

Seabra, J. C., Costa, M. A., & Lucena, M. M. (2016). IoT based intelligent system for fault detection and diagnosis in domestic appliances. In *2016 IEEE 6th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)* (pp. 205–208). IEEE. https://doi.org/10.1109/ICCE-Berlin.2016.7684756

Simske, S. J. (2013). *Meta-algorithmics: patterns for robust, low cost, high quality systems*. John Wiley & Sons.

Sinha, S., & Lee, Y. M. (2024). Challenges with developing and deploying AI models and applications in industrial systems. *Discover Artificial Intelligence, 4*, 55. https://doi.org/10.1007/s44163-024-00151-2

Smart, W. D., Grimm, C. M., & Hartzog, W. (2021). An education theory of fault for autonomous systems. *Notre Dame J. on Emerging Tech.*, *2*, 33.

Smith, D. J. (2021). *Reliability, maintainability and risk: practical methods for engineers*. Butterworth-Heinemann.

Stergiopoulos, G., Kotzanikolaou, P., Theocharidou, M., Lykou, G., & Gritzalis, D. (2016). Time-based critical infrastructure dependency analysis for large-scale and cross-sectoral failures. *International Journal of Critical Infrastructure Protection, 12*, 46–60. https://doi.org/10.1016/j.ijcip.2015.12.002

Wilson, M., & Goffnett, S. (2022). Reverse logistics: Understanding end-of-life product management. *Business Horizons*, *65*(5), 643-655.

Xu, G., Liu, M., Jiang, Z., Shen, W., & Huang, C. (2019). Online fault diagnosis method based on transfer convolutional neural networks. *IEEE Transactions on Instrumentation and Measurement*, *69*(2), 509-520.

BIOGRAPHIES

Ryan J Aalund is pursuing a D. Eng. in Systems Engineering at Colorado State University. He holds a B.S. in Electronics Engineering Technology, an M.S. in Electrical Engineering from DeVry University, and an M.S. in Reliability Engineering from the University of Maryland. He serves as the Director of Hardware Reliability and Quality at Samsara. His research focuses on system reliability in IoT and connected technologies.

Meenusree Rajapandian currently works as an Applied Scientist at Samsara Inc. She holds an M.S. in Industrial Engineering from Purdue University and a B.E. in Electrical Electronics Engineering from Anna University. Her research centers on applying AI to develop autonomous closed loop decision-making systems for complex industrial operations.

Vincent Philip Paglioni was born in Kennesaw, Georgia, USA in 1993. He received his B.S. in Nuclear and Radiological Engineering from Georgia Institute of Technology in 2017. He received his M.S. and Ph.D. in Reliability Engineering from the University of Maryland, College Park in 2022 and 2023, respectively.

From 2017 to 2019, he was a Nuclear Test Engineer for the Department of Navy at Portsmouth Naval Shipyard. Since 2023, he has been an Assistant Professor with the Department of Systems Engineering at Colorado State University in Fort Collins, Colorado. His research interests include human reliability analysis (HRA) for nuclear power, risk and reliability analysis for complex systems, resiliency engineering, and the intersection of ethics with risk assessment.