ALARM MANAGEMENT OF VENTILATOR USING MACHINE LEARNING TECHNIQUES

M.Tech. Thesis

By

MRUTYUNJAY CHINCHOLE

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2025

Alarm Management of Ventilator using Machine Learning Techniques

A THESIS

Submitted in partial fulfillment of the requirements for the award of the degree
of
Master of Technology

by

MRUTYUNJAY CHINCHOLE

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
MAY 2025

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Alarm Management of Ventilator using Machine Learning Techniques** in the partial fulfillment of the requirements for the award of the degree of **Master of Technology** and submitted in the **Department of Mechanical Engineering, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from July 2023 to May 2025 under the supervision of Prof. Bhupesh Kumar Lad .

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

07-06-25

Signature of the student with date
MRUTYUNJAY CHINCHOLE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of the Supervisor of
M.Tech. thesis (with date)

PROF. BHUPESH KUMAR LAD

MRUTYUNJAY CHINCHOLE has successfully given his M.Tech. Oral Examination held on **26th May 2025**.

09/06/2025

Signature of Supervisor of M.Tech. thesis
Date: 07/06/25

Convener, DPGC
Date:

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who supported me throughout the course of this thesis.

First and foremost, I am deeply thankful to my guide, **Prof. Bhupesh Kumar Lad**, for their unwavering support, insightful feedback, and continuous encouragement. Their guidance has been instrumental in shaping both the direction and quality of this research.

I would like to extend heartfelt thanks to my colleagues, **Miss Arzoo** and **Mr. Md Sibgatullah Ahmad**, for their valuable contributions, collaborative efforts, and assistance during critical phases of the project. I would also like to thank **Dr. Nidhi Pathak** for her valuable feedback and support.

I am also grateful to **Indian Institute of Technology Indore** and **Charak centre for Digital Healthcare** for providing the essential facilities, resources, and an encouraging research environment. A special note of appreciation goes to **Dr. Deepak Agrawal,** whose clinical insights and expert suggestions significantly enhanced the relevance and depth of this work.

A very special and personal thank you to my father, **Dr. Mahadev Chinchole**, whose dual role as a medical professional and a guiding parent greatly benefited this research. His insights, critical observations, and continuous support were invaluable, both academically and emotionally.

I would also like to acknowledge the contribution of all the **technical experts, lab technicians, and support staff** from both the college and associated hospital settings. Their cooperation and timely assistance were crucial in data collection, equipment handling, and ensuring the smooth progress of various experimental and practical components of this work.

I am sincerely thankful to my friends for their constant motivation and support, and most importantly, to my family for their unconditional love, patience, and belief in me throughout this journey.

This work would not have been possible without the contribution and support of all the above individuals and institutions.

Thank you all.

Dedicated to my beloved Grandparents

Abstract

Digital transformation in healthcare has ushered in an era of interconnected, data-driven clinical environments, enabling continuous monitoring and analysis of patient status. As care complexity increases, smart decision-support systems have emerged to assist medical staff in interpreting vast streams of physiological data, addressing the growing need for timely and accurate insights at the point of care.

Alarm fatigue poses a huge problem in the healthcare sector, which critically undermines patient safety by desensitizing clinicians to life-threatening alarms, increasing the risk of missed or delayed responses. Its pervasive cognitive overload and workflow disruptions also contribute to clinician burnout and medical errors.

This work emphasizes the critical role of Alarm Management Systems within Intensive Care Units, where alarm overload and false alarms contribute to clinician fatigue and potential safety risks. The novelty of this project lies in the fact that no similar solution exists, we generated a dataset by simulating patient conditions and capturing sensor data for several disease states both with and without induced system or patient faults.

The proposed system is designed to automatically flag critical alarms the moment they arise and to generate context-aware recommendations for both medical and technical staff, thereby streamlining response workflows. By integrating machine-learning algorithms with real-time sensor inputs, our approach prioritizes actionable

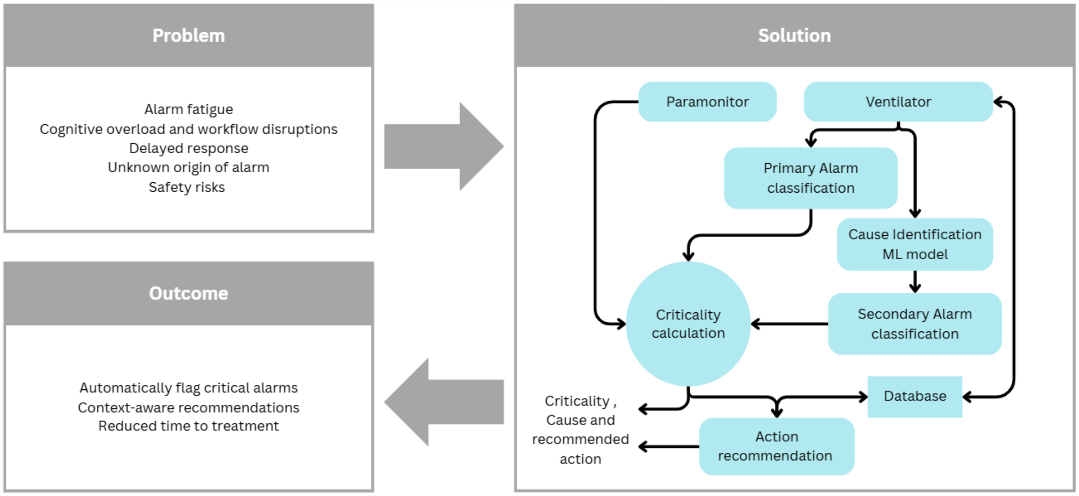

Problem
Alarm fatigue
Cognitive overload and workflow disruptions
Delayed response
Unknown origin of alarm
Safety risks
Solution
Paramonitor
Ventilator
Primary Alarm classification
Cause Identification ML model
Secondary Alarm classification
Criticality calculation
Database
Action recommendation
Criticality , Cause and recommended action
Outcome
Automatically flag critical alarms
Context-aware recommendations
Reduced time to treatment

Fig Project Overview

alerts and suggests evidence-based interventions, reducing unnecessary disruptions while ensuring rapid attention to life-threatening events.

The framework calculates the real-time criticality of an alarm based on patient condition (through vitals) and the type of alarm generated (which is identified using ML models). This generates an output that indicates whether the alarm is critical or not. Further mapping this criticality, alarm type, and cause of the alarm leads to suggesting responsive action for that specific alarm. This framework aims to enhance patient safety, optimize staff efficiency, and demonstrate how advanced alarm management can form a cornerstone of next-generation ICU care.

Future work will focus on translating our proof-of-concept into a fully integrated software solution capable of seamless deployment within clinical environments. Concurrently partnerships with healthcare institutions to collect and curate large-scale, real-world alarm datasets across diverse patient populations and device configurations. This expanded data corpus will enable us to retrain and validate our machine-learning models, enhancing their robustness, generalizability, and resilience to noise and variability in clinical practice. Ultimately, these efforts aim to ensure that our Alarm Management System not only performs effectively in controlled simulations but also delivers reliable, high-impact support in live ICU settings.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI	Artificial Intelligence
ARIMA	Autoregressive Integrated Moving Average
BIS	Bispectral Index
CCU	Coronary (or Cardiac) Care Unit
CO_2	Carbon Dioxide
CPR	Cardiopulmonary Resuscitation
ECG	Electrocardiogram
ED	Emergency Department
EEG	Electroencephalogram
EHR	Electronic Health Record
$EtCO_2$	End-Tidal Carbon Dioxide
HR	Heart Rate
IBP	Invasive Blood Pressure
ICU	Intensive Care Unit
LSTM	Long Short-Term Memory Model
ML	Machine Learning
NIBP	Non-Invasive Blood Pressure
OR	Operating Room
PEEP	Positive End Expiratory Pressure
RR	Respiratory Rate
SpO_2	Peripheral Capillary Oxygen Saturation
VAP	Ventilator Associated Pneumonia

Chapter 1: Introduction

1.1. Industry 4.0

Industry 4.0 represents the shift from conventional manufacturing toward highly digitalized, data-driven production systems. Originating from Germany's "High-Tech Strategy 2020,[1]" it focuses on merging cyber-physical systems, the Internet of Things, and seamless data exchange to create "smart factories." In these environments, equipment, sensors, and operators interact instantaneously to enable self-adjusting, flexible workflows. The overarching objectives are to boost efficiency, minimize waste, shorten lead times, and support individualized mass production—all while maintaining high quality.

The following core technologies form the foundation of Industry 4.0:

1. Cyber-Physical Systems (CPS)

   CPS integrates computing algorithms with physical machinery, so devices can both monitor and influence their surroundings. In manufacturing, embedded microprocessors, sensors, and actuators cooperate to supervise and manage every production step. Characteristically, CPS implement:

   - In-Line Monitoring and Control: Sensors collect real-time data (e.g., temperature, vibration), and actuators autonomously modify machine settings (e.g., spindle speed, conveyor rate).

   - Immediate Feedback Loops: Local (edge) processing enables rapid decisions- safety interlocks or quality adjustments without relying on a distant server.

   - Distributed Intelligence: Instead of a single centralized controller, decision-making is spread across multiple nodes, allowing machines to anticipate faults, predict maintenance needs, and collaborate with nearby devices [2] [3].

2. Industrial Internet of Things (IIoT)

   IIoT extends IoT principles to industrial equipment, using reliable protocols (such as OPC UA and MQTT) to interconnect machinery, production lines, storage systems, and enterprise software. Its key traits are:

   - Broad Connectivity: Every sensor, robot, and even workpiece has a unique digital identifier, continuously exchanging status updates, performance metrics, and maintenance alerts.
   - Edge and Fog Computing: Rather than transmitting all data to a central cloud, preliminary analytics (filtering, aggregation, anomaly detection) occur near the machine ("edge") or in intermediate "fog" nodes, reducing latency and conserving bandwidth.
   - Scalability and Interoperability: IIoT architectures scale to accommodate thousands of devices and support equipment from various manufacturers, ensuring unified communication across heterogeneous systems [3] [4]

3. Big Data Analytics and Data Management

   Industrial environments generate vast volumes of streaming data from PLCs, robots, and vision systems. Advanced analytics (machine learning, data mining, predictive models) are essential for deriving actionable insights:

   - Real-Time Data Ingestion: High-speed pipelines ingest continuous data streams into time-series databases or data lakes for storage and analysis.
   - Predictive and Prescriptive Models: Machine learning algorithms forecast equipment failures (predictive maintenance), optimize operational parameters (e.g., temperature, feed rate), and simulate "what-if" scenarios to inform production decisions.

4. Digital Twins

   Virtual replicas of physical assets receive live data feeds, enabling engineers to test process changes or equipment modifications in a simulated environment before implementing them on the factory floor [4].

5. Cloud Computing and Software-as-a-Service (SaaS)

   Cloud platforms supply on-demand computing power, storage, and industrial software without requiring extensive local infrastructure:

   - Elastic Resources: Manufacturers can provision additional virtual machines or containers to handle peak workloads such as large-scale simulations or analytics during intensive production runs.
   - Centralized Data Repositories: Data from multiple plants is aggregated in the cloud, facilitating enterprise-wide benchmarking, cross-site analysis, and shared analytics.
   - SaaS Applications: Cloud-hosted ERP, MES, quality management, and supply chain solutions can be deployed rapidly and updated continuously, ensuring the latest features and security patches [5].

6. Artificial Intelligence (AI) and Machine Learning (ML)

   AI and ML leverage historical and real-time data to detect patterns, make predictions, and automate complex tasks:

   - Computer Vision: Deep learning-based image recognition inspects components on production lines with higher accuracy than manual inspections- identifying defects, alignment issues, or surface contaminants in milliseconds.
   - Anomaly Detection and Predictive Maintenance: Unsupervised ML models detect deviations from normal operating parameters such as abnormal vibration frequencies alerting operators before machine breakdowns occur.

- Reinforcement Learning for Process Control: Advanced AI agents continuously tune process variables (e.g., pressure, feed rate) to optimize yield or energy usage, learning optimal operating policies over time [2], [4].

7. Additive Manufacturing (3D Printing)

   Additive manufacturing constructs parts layer by layer from digital models, enabling rapid prototyping and production of complex geometries. In Industry 4.0:

   - Design Optimization: Generative design tools and topology optimization software produce lightweight, structurally efficient components that conventional machining cannot easily fabricate.
   - Distributed Production Models: Digital part files are transmitted to regional "micro-factories" or service bureaus, reducing shipping times and carrying minimal inventory.
   - Advanced Materials: Novel metal powders, polymers, and composite materials feed next-generation printers, supporting the on-demand manufacture of fixtures, tooling, and end-use parts including biomedical implants [5].

8. Augmented Reality (AR) and Virtual Reality (VR)

   AR overlays digital instructions onto the physical world, while VR immerses users in virtual environments for planning, training, or simulation:

   - Hands-Free Work Instructions: AR headsets (e.g., HoloLens) display step-by-step guidance, 3D component models, and safety warnings in the operator's field of view reducing errors and accelerating changeovers.
   - Remote Expert Collaboration: An on-site technician can live-stream equipment views to an off-site specialist, who annotates the AR feed in real time to guide maintenance or troubleshooting.

- VR Layout and Ergonomic Simulation: Engineers can virtually walk through proposed production line configurations, test ergonomic considerations, and verify safety clearances before physical installation [3].

9. Horizontal and Vertical System Integration

    - Horizontal Integration: Entails seamless data flow across the entire supply chain—connecting suppliers, manufacturing, logistics, and customers. Cloud-based supply chain platforms can automatically reorder parts, track shipments in real time, and synchronize production schedules to actual demand.

    - Vertical Integration: Breaks down traditional automation silos (PLC, SCADA/MES, ERP) by implementing unified data models (e.g., OPC UA, RAMI 4.0) so that shop-floor controllers, edge gateways, MES, and ERP systems share a synchronized digital view of operations [2], [4].

10. Cybersecurity in Industrial Control Systems (ICS)

    As manufacturing environments become highly networked, built-in cybersecurity is crucial:

    - Zero Trust Principles: Every device, user, and application is untrusted by default; strict authentication, encrypted communications, and least-privilege access controls are enforced throughout the network.

    - Network Segmentation: Separating critical control networks (PLCs, SCADA) from corporate IT networks and applying granular firewall rules minimizes the attack surface.

    - Continuous Monitoring and Threat Hunting: Real-time anomaly detection spotting unusual command sequences or data exfiltration attempts is embedded in ICS event logs and SIEM systems to detect and contain threats quickly [4], [5].

11. Collaborative Robotics (Cobots) and Autonomous Mobile Robots (AMRs)

Unlike traditional industrial robots confined behind safety fences, cobots and AMRs operate safely alongside human workers:

- Force-Sensing and Safe Human–Robot Interaction: Cobots incorporate torque and force sensors to detect unexpected collisions and halt motion immediately, preventing injuries.
- Dynamic Navigation and Path Planning: AMRs use LiDAR, cameras, and floor markers to autonomously navigate complex factory or warehouse layouts, avoiding obstacles and optimizing delivery routes for parts or finished goods.
- Plug-and-Play Deployment: Cobots often require minimal programming; operators can manually guide the robot through tasks, and it learns by demonstration [2], [3].

Example of an Industry 4.0–Enabled Facility

1. Smart Sensors and IIoT Gateways continuously collect vibration, temperature, and energy data from CNC mills, presses, and injection-molding machines.
2. Edge Computing Nodes perform local analytics—detecting anomalies and autonomously adjusting spindle speeds to prevent tool wear.
3. Cloud-Hosted Analytics aggregate data from multiple plants to benchmark energy consumption and identify performance outliers.
4. Robust Cybersecurity ensures only authenticated devices can update critical controllers, preventing unauthorized access.
5. AR Glasses overlay 3D models and torque specifications onto a technician's field of view during maintenance.
6. Cobots switch between tasks—such as inspection and palletizing—automatically as production orders change in real time.

7. Additive Manufacturing Cells fabricate customized jigs on demand: an engineer in Detroit sends a CAD file to a Berlin print cell, which produces the fixture within hours.

By integrating CPS, IIoT, big data analytics, AI/ML, cloud/SaaS, additive manufacturing, AR/VR, horizontal/vertical integration, cybersecurity, and advanced robotics, Industry 4.0 transforms static, siloed factories into responsive, self-optimizing ecosystems capable of anticipating changes, self-healing from disruptions, and delivering higher value at lower cost and risk.

1.2. Digital healthcare and Industry 4.0

Bridging the gap between conventional healthcare and its digital counterpart requires an ecosystem of interoperable, data-driven technologies that enhance access, efficiency, and personalization of care. At its foundation lie Electronic Medical/Health Record systems (EMR/EHR), which centralize patient histories, diagnostics, and treatment plans, enabling seamless information flow across departments and care teams [6]. Layered atop this data backbone are Internet-of-Things (IoT) devices and wearable sensors that continuously capture vital signs and other physiological metrics, feeding real-time streams into cloud platforms for aggregation and long-term storage. Artificial Intelligence (AI) and Machine Learning (ML) then transform these massive datasets into actionable insights—powering predictive diagnostics, risk stratification, and decision-support tools that alert clinicians to early signs of deterioration or recommend personalized treatment pathways [7]. Telemedicine and mobile-health (mHealth) applications extend the clinical reach beyond hospital walls, while 5G and edge-computing architectures ensure low-latency, reliable connectivity for remote monitoring and procedural guidance [8]. Blockchain and advanced cybersecurity protocols safeguard data integrity and patient privacy, fostering trust in digital systems. Together, these Industry 4.0–inspired components—from EMR/EHR and big-data analytics to AI/ML, IoT, telehealth, and secure cloud infrastructures—form a cohesive digital healthcare framework [9]. By integrating technologies at every layer of care delivery, we can move from reactive, episodic interventions to proactive, continuous, and patient-centric management, truly bridging the divide between traditional practice and next-generation healthcare.

Digital healthcare utilizes digital technologies to enhance healthcare delivery, making it more accessible, efficient, and cost-effective. It encompasses a wide range of tools, including telemedicine, mobile health apps, electronic health records, and more. These technologies aim to transform healthcare by empowering patients, improving care quality, and expanding access, especially in remote areas.

1.3. AI and ML Introduction

Artificial Intelligence (**AI**) broadly refers to computational systems designed to emulate human-like cognitive functions—such as perception, reasoning, problem-solving, and decision-making. AI encompasses a wide range of techniques (rule-based systems, expert systems, evolutionary algorithms, and biologically inspired approaches) meant to automate tasks that traditionally require human intelligence [10], [11], [12], [13], [14].

Machine Learning (**ML**) is a subset of AI focused specifically on algorithms that enable computers to "learn" from data rather than rely on hand-coded rules. In ML, models detect patterns or regularities in examples (training data) and generalize these patterns to make predictions or decisions on new, unseen inputs [10], [11], [12], [13], [14].

Categories of Machine Learning [10], [11], [12], [13], [14]

Machine learning can be broadly categorized into several paradigms based on how models interact with data and the learning signals they receive. In its simplest form, **Supervised Learning** involves training models on labelled datasets—each example in the training set pairs an input vector with a "ground-truth" output. The model's objective is to learn a mapping from inputs to outputs by minimizing the discrepancy between its predictions and the known labels. Within supervised learning, two primary tasks emerge**. Classification** aims to predict discrete categories. In **a binary classification** scenario, the model must distinguish between two classes (for example, spam versus not-spam or malignant versus benign). In a **multiclass classification** setting, there are more than two possible labels—common examples include recognizing handwritten digits (0 through 9) or determining sentiment (positive, neutral, or negative). **Regression**, by contrast, focuses on forecasting continuous numerical values. A typical instance is **linear regression**, where one predicts house prices based on features such as square footage or number of bedrooms. When the relationship between features and the target variable is nonlinear, methods like **polynomial regression** or other nonlinear regression techniques can be applied (for example, modelling population growth as a nonlinear function of time).

In contrast, **Unsupervised Learning** deals with unlabelled data and tasks the model with uncovering inherent structures, groupings, or low-dimensional embeddings without explicit "correct answers." A major subclass is **clustering**, where the algorithm partitions data points into groups of similar examples. One widely used method is **k-Means Clustering**, which iteratively assigns points to *k* clusters to minimize within-cluster variance. Another is **hierarchical clustering**, which constructs a tree of clusters either by successively merging smaller clusters (agglomerative) or by dividing larger ones (divisive). **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) forms clusters based on regions of high point density and can identify arbitrarily shaped clusters as well as outliers. A second key unsupervised task is **dimensionality reduction**, which seeks to compress high-dimensional inputs into a smaller set of latent features while retaining as much "information" as possible. **Principal Component Analysis** (**PCA**) is a linear technique that projects data onto orthogonal axes of maximum variance, whereas **t-Distributed Stochastic Neighbour Embedding** (t-SNE) is a nonlinear method especially well-suited for visualizing high-dimensional data in two or three dimensions. **Autoencoders**, which are neural-network-based encoders–decoders, learn compressed representations by training the network to reconstruct its input. Lastly, **Association Rule Learning** discovers "if-then" relationships between variables—classic algorithms include **Apriori**, which finds frequent itemsets in transactional data (for example, market-basket analysis), and **Eclat**, which uses depth-first search to enumerate itemsets.

Reinforcement Learning (**RL**) represents a third paradigm in which an agent interacts with an environment in discrete time steps: at each step, the agent selects an action, receives a reward (or penalty), and transitions to a new state. The goal is to learn a policy—a mapping from states to actions—that maximizes cumulative reward over time, balancing exploration (trying new actions) against exploitation (leveraging known rewarding actions). Within RL, model-**free methods** learn value functions or policies directly from experience without constructing an explicit model of environment dynamics. Q-**Learning** is a prototypical model-free algorithm that estimates the action-value function Q(s,a) via temporal-difference updates:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \mathrm{max} Q(s',a') - Q(s,a)]$$

where α is the learning rate, r is the received reward, γ is the discount factor, and s′ is the next state. **SARSA** (State-Action-Reward-State-Action) is an on-policy variant that updates Q based on the action actually taken at the next state. **Policy-gradient methods** directly parameterize the policy π(a|s;θ) and adjust the parameters θ by ascending the gradient of expected reward; examples include the classic **REINFORCE** algorithm. **Actor**–**Critic** approaches combine a policy network (actor) with a value network (critic) so that the critic's value estimates reduce variance in the policy gradient updates. In **model-based RL**, the agent explicitly builds or is provided with a model of environment dynamics (i.e., transition probabilities and reward functions) and uses planning techniques—such as dynamic programming—to compute optimal actions. In modern settings, **Deep Reinforcement Learning** leverages deep neural networks as function approximators, enabling RL to scale to high-dimensional inputs like raw images. Notable examples include the **Deep** Q **Network** (DQN), which integrates Q-learning with convolutional neural networks to play Atari games from pixel inputs; **Deep Deterministic Policy Gradient** (**DDPG**), which extends actor–critic approaches to continuous action spaces; and policy-gradient refinements such as **Proximal Policy Optimization** (**PPO**) and **Trust Region Policy Optimization** (**TRPO**), which enforce constraints on policy updates to stabilize training.

Between supervised and unsupervised extremes lies **Semi-Supervised** Learning, where a small amount of labelled data is augmented by a large pool of unlabelled examples. The objective is to exploit the unlabelled set to learn the underlying data distribution more accurately, which is especially beneficial when labelling is expensive or time-consuming. One approach is **self-training**, where a classifier trained on labelled data is used to generate pseudo-labels for unlabelled examples; high-confidence predictions are then added to the labelled set in an iterative fashion. **Co-training** trains two or more models on different "views" or feature subsets of the data; each model labels the unlabelled examples that the other models then use for training. **Graph-based methods** build a similarity graph among samples (nodes) and propagate label information along edges to infer unknown labels. Meanwhile, **semi-supervised generative models**, such as variational autoencoders with an auxiliary classification objective, jointly optimize a generative component (modelling p(x)p(x)p(x)) and a discriminative component (modelling p(y|x)p(y \mid x)p(y|x)) to leverage both labelled and unlabelled data.

Self-Supervised Learning is an emerging paradigm in which the model derives supervisory signals automatically from the data itself, setting up "pretext tasks" whose solution requires learning useful representations. In computer vision, one example is **context prediction**, where a network attempts to predict the relative position of one image patch given another. In natural language processing, **masked language modelling** (as in BERT) randomly hides a subset of tokens and trains the network to predict the masked words from their surrounding context. Another powerful framework is **contrastive learning**, which constructs positive pairs (two augmented views of the same input) and negative pairs (views from different inputs) and trains the encoder to pull together representations of positives while pushing apart negatives. The representations learned by solving these pretext tasks can then be fine-tuned for downstream objectives such as classification or detection.

Deep Learning: A Specialization of Machine Learning

Deep Learning (DL) is a subfield of machine learning that focuses on using multi-layer neural networks—commonly referred to as deep neural networks—to automatically learn hierarchical feature representations. Although DL methods can be applied under any of the paradigms above (supervised, unsupervised, reinforcement, self-supervised), the hallmark of DL is its capacity to handle very high-dimensional and unstructured data by learning multiple levels of abstraction. A basic architecture is the **Feedforward Neural Network (FNN)**, in which layers of neurons are stacked such that each layer's outputs serve as inputs to the next, with no cycles or feedback connections. FNNs are versatile for generic tasks in classification and regression.

For data with spatial structure—such as images or any grid-like inputs—**Convolutional Neural Networks (CNNs)** are the preferred choice. CNNs use learnable convolutional filters and pooling layers to exploit local correlations (e.g., edges in images), drastically reducing the number of parameters compared to fully connected networks. CNN architectures have led to state-of-the-art performance in image classification, object detection, semantic segmentation, and other computer-vision tasks. When the data are sequential—such as sentences, time series, or speech—**Recurrent Neural Networks (RNNs)** offer mechanisms to process inputs in a temporal manner by maintaining hidden states across time steps. To address the vanishing/exploding gradient problems of vanilla RNNs, specialized RNN variants such as **Long Short-Term Memory (LSTM)** and **Gated Recurrent Unit (GRU)** were developed; these architectures employ gating mechanisms that make it easier to capture long-range dependencies.

Graph-structured data—where instances are nodes connected by arbitrary edges—are effectively modelled by **Graph Neural Networks (GNNs)**. A GNN propagates information along edges to learn node-level or graph-level representations; this is useful in social-network analysis, molecular property prediction, recommendations, and any domain where relationships between entities matter. More recently, **Transformer models**, built around self-attention mechanisms, have revolutionized sequence modelling. Initially introduced for machine translation, transformers like

BERT and the **GPT** series capture global dependencies by computing pairwise attention scores, making them highly effective for language understanding. Vision Transformers (ViTs) have adapted the same architecture to image patches, showing competitive results with CNNs in vision tasks. Deep Learning's flexibility allows these architectures to be deployed in fully supervised settings, for unsupervised representation learning (e.g., autoencoders or contrastive methods), and within reinforcement-learning frameworks (e.g., DQN's CNN backbone).

Additional ML Paradigms and Specialized Subtypes

Beyond the core paradigms of supervised, unsupervised, and reinforcement learning, several other approaches have gained traction in specific applications or to overcome particular challenges:

- **Transfer Learning** refers to techniques that leverage knowledge obtained from one task or domain and apply it to another related task. A canonical example is taking a CNN pretrained on ImageNet—a dataset of millions of labeled images—and fine-tuning it for a specialized task like medical image classification. Transfer learning dramatically reduces training time and the amount of labeled data required in the target domain.
- **Online Learning (Incremental Learning)** trains or updates models sequentially as new data arrive, rather than assuming access to a fixed, static dataset. This is crucial for streaming applications—such as real-time anomaly detection in network traffic—where patterns can shift over time, and the model must adapt continuously without retraining from scratch.
- **Federated Learning** is a decentralized training paradigm designed to preserve data privacy. Instead of sending raw data to a central server, each edge device (e.g., smartphone, IoT sensor) trains a local model on its private data and transmits only model updates (such as gradients or weight differences) to a central aggregator. The server then aggregates these updates (for example, via weighted averaging) to form a global model, which is sent back to each device. Federated learning has proven valuable in healthcare, finance, and mobile applications where data sensitivity is paramount.

- **Multi-Task Learning** trains a single model to perform multiple related tasks simultaneously by sharing representations among tasks. For instance, in computer vision, one might train a network to simultaneously predict object bounding boxes and classify object categories. By jointly optimizing for related objectives, multi-task learning often yields better generalization than training separate models for each task.
- **Ensemble Learning** improves predictive performance and robustness by combining multiple base learners. In **bagging** (Bootstrap Aggregating), several models are trained on different bootstrap samples of the data, and their outputs are averaged (or voted) to produce a final prediction; the Random Forest algorithm is a prime example of a bagging ensemble of decision trees. In **boosting**, base models are trained sequentially, with each new model focusing on correcting errors made by its predecessors; prominent algorithms include AdaBoost, Gradient Boosting Machines, and XGBoost. **Stacking** (or stacked generalization) trains multiple base models in parallel and then uses a "meta-learner" to combine their predictions optimally—often yielding further gains over bagging or boosting alone.

Collectively, these paradigms and subtypes illustrate the richness of modern machine learning: from traditional supervised and unsupervised techniques to advanced online, federated, and multitask frameworks. Deep learning architectures—such as CNNs, RNNs, GNNs, and transformers—can be applied within many of these paradigms, enabling practitioners to tackle a diverse array of real-world data modalities, from images and text to graphs and streaming signals.

Cross validation and Hyperparameter tuning:

Cross-validation is a technique used in machine learning to evaluate how well a model will perform on unseen data by systematically splitting the available dataset into multiple train/validation subsets. Instead of relying on a single train/test split—which can produce misleading performance estimates due to random data partitioning—k-fold cross-validation divides the data into *k* equally sized "folds." The model trains on *k*–1 folds and validates on the remaining fold, repeating this process *k* times so that every sample serves as validation exactly once. By averaging the performance metrics across all folds, one obtains a more reliable estimate of the model's true generalization ability [10], [11].

Beyond the basic k-fold approach, several variants exist to address specific data challenges. Stratified k-fold ensures that each fold maintains the same class proportions as the overall dataset, preventing skewed validation results when dealing with imbalanced classes. Leave-one-out (LOO) cross-validation uses each individual sample as its own validation set, which maximizes training data but is computationally intensive and can yield high-variance estimates. For time-series or sequential data, forward-chaining (time-based) cross-validation respects temporal order by training on past observations and validating on future ones. Group or block cross-validation keeps related samples (e.g., all data from the same subject or device) together in either the training or validation fold to avoid data leakage [11], [15].

Implementing cross-validation properly requires that all preprocessing steps—such as scaling, encoding, or feature selection—be fit only on the training folds and then applied to the validation fold. This ensures no information from the validation set "leaks" into training. Cross-validation is often combined with hyperparameter tuning (e.g., grid search), where each candidate hyperparameter configuration is evaluated via cross-validation and the one with the best average validation performance is selected. Overall, cross-validation provides a robust framework to prevent overfitting, guide model selection, and generate stable performance estimates, at the cost of increased computational effort for retraining across multiple folds [10], [15].

1.4. Classification models applied in this work

Logistic Regression is a simple yet powerful linear classification algorithm that is widely used for both binary and multiclass classification tasks. In multiclass problems, it uses the softmax function to estimate the probability that a given input belongs to each class. Logistic regression assumes a linear relationship between input features and the log-odds of the classes, making it very interpretable and fast to train.

It is commonly used as a baseline model because of its simplicity and effectiveness on linearly separable data [16].

Decision Tree Classifier is a non-parametric supervised learning method used for classification. It works by recursively splitting the dataset into subsets based on feature values that minimize impurity metrics like Gini index or entropy. The resulting tree structure allows for intuitive decision-making and interpretation. Decision trees can capture non-linear relationships and interactions between variables, making them useful for a wide range of problems involving tabular data [17].

Random Forest is an ensemble learning method that builds multiple decision trees and merges their outputs to improve accuracy and reduce overfitting. It does this by training each tree on a random subset of data (bagging) and then aggregating their results, usually via majority voting in classification tasks[18]. Random Forest is more robust than a single decision tree and performs well on high-dimensional data, making it a go-to model for many real-world structured datasets.

k-Nearest Neighbors (kNN) is a simple and intuitive classification algorithm that assigns a class to a new data point based on the majority vote of its k nearest neighbors in the feature space. It is a lazy learner, meaning it doesn't learn a model during training but stores the dataset and uses it for prediction. Although kNN can work well for datasets with clear class boundaries, it can be computationally expensive for large datasets and sensitive to feature scaling [19].

MLPClassifier, or Multi-Layer Perceptron, is a type of feedforward artificial neural network. It consists of input, hidden, and output layers with non-linear activation functions, allowing it to capture complex patterns in data. MLP is trained using backpropagation and gradient descent. While it requires more computational resources and tuning than simpler models, it is very flexible and can approximate any decision boundary given enough data and capacity [13].

Support Vector Machine (SVM) is a robust classifier that finds the optimal hyperplane separating different classes with the maximum margin. In multiclass problems, SVM uses strategies like one-vs-rest or one-vs-one, and it can handle non-linear data by applying the kernel trick (e.g., RBF kernel) [20]. SVMs are especially effective in high-dimensional spaces and are known for their generalization ability, although they can be slow on large datasets.

XGBoost (Extreme Gradient Boosting) is a high-performance ensemble learning technique based on gradient boosting. It builds trees sequentially, where each tree tries to correct the errors made by the previous one. It uses regularization to reduce overfitting and supports advanced features like missing value handling and tree pruning. XGBoost is highly efficient and accurate, making it a favorite in machine learning competitions and practical applications [21].

CatBoost is another gradient boosting framework that stands out for its native support for categorical variables. It uses a technique called ordered boosting and symmetric trees to achieve faster convergence and better generalization. CatBoost is designed to work well with minimal preprocessing and is optimized for both speed and accuracy. It is particularly useful when working with mixed datasets containing both numerical and categorical features [22].

LightGBM is a gradient boosting framework developed by Microsoft, known for its speed and efficiency. It uses a histogram-based algorithm to bucket continuous features, allowing it to train faster and use less memory. LightGBM also uses a leaf-wise tree growth strategy, which often leads to better accuracy compared to level-wise approaches. It is suitable for large-scale datasets and high-dimensional features and supports native multiclass classification [23].

1.5. Intensive Care Unit (ICU)

An ICU is a specialized hospital area focused on continuously managing patients with severe, potentially life-threatening conditions. Individuals in the ICU need constant monitoring because their vital signs such as heart rate, blood pressure, respiratory function, and oxygen saturation can fluctuate suddenly and significantly.

ICUs operate 24/7 with a collaborative, multidisciplinary team that includes intensivists (physicians trained in critical care), critical-care nurses (who often manage one or two patients at a time), respiratory therapists, pharmacists, and other experts. This team works together to interpret detailed patient data, respond quickly to changing situations, and adjust complex treatments without delay.

Patients often rely on advanced life support devices in the ICU like mechanical ventilators to assist or replace breathing, infusion pumps for precise medication and fluid delivery, and continuous renal replacement machines to support kidney function. Bedside monitors continuously display a range of metrics like ECG readings, invasive blood pressures, oxygen levels with alarms that alert staff to any worrisome shifts.

The ICU's main objective is to stabilize critically ill patients, correct dangerous imbalances (such as severe infections or respiratory failure), and prevent complications like secondary organ injury or hospital-acquired infections. Once a patient's condition is stable and they can maintain vital functions with less intensive support, they are transitioned often to a step-down unit or general ward where they continue to recover under less rigorous monitoring.

1.5.1. Ventilator

Ventilation (Physiology)- The movement of air between the environment and the lungs via inhalation and exhalation.

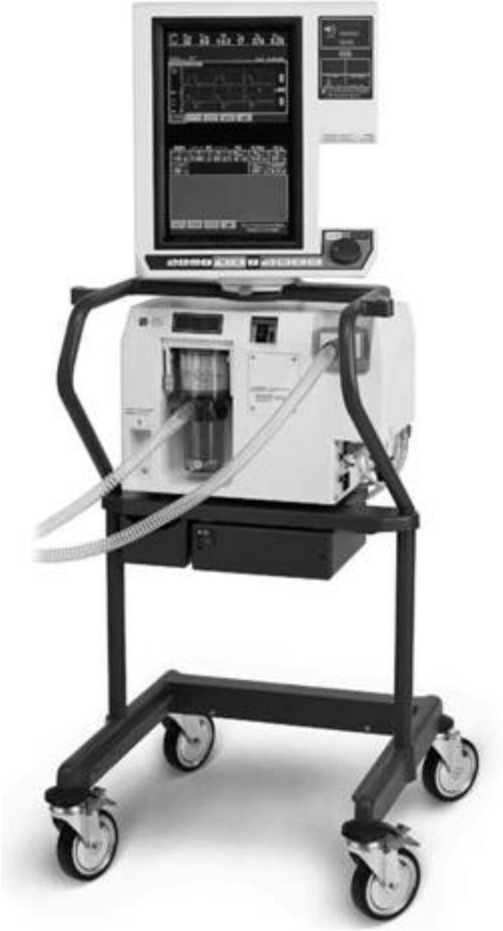


Figure 1 PB 840 ventilator

Mechanical Ventilation- In medicine, using artificial methods to assist breathing.

Ventilator- A machine designed to move breathable air into and out of the lungs.

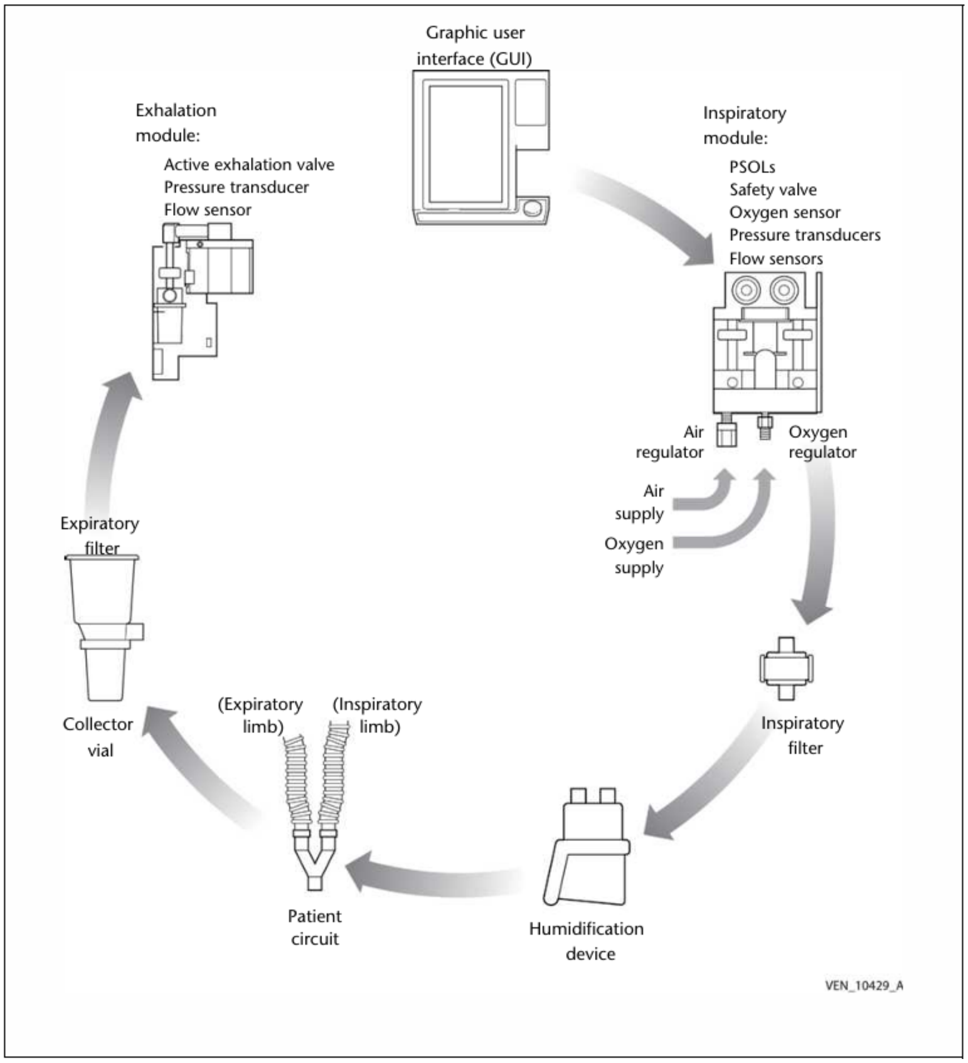


Figure 2Ventilator System block diagram

The diagram illustrates the components and airflow path of a mechanical ventilator system: starting at the graphic user interface (GUI), gas (air and oxygen) is regulated and conditioned through the inspiratory module (including pressure/flow sensors and filters), then passed through a humidification device into the patient circuit (inspiratory limb). Exhaled gas returns via the expiratory limb, passes through an expiratory filter and collection vial, and is finally managed by the exhalation module's active valve and sensors before completing the cycle [24].

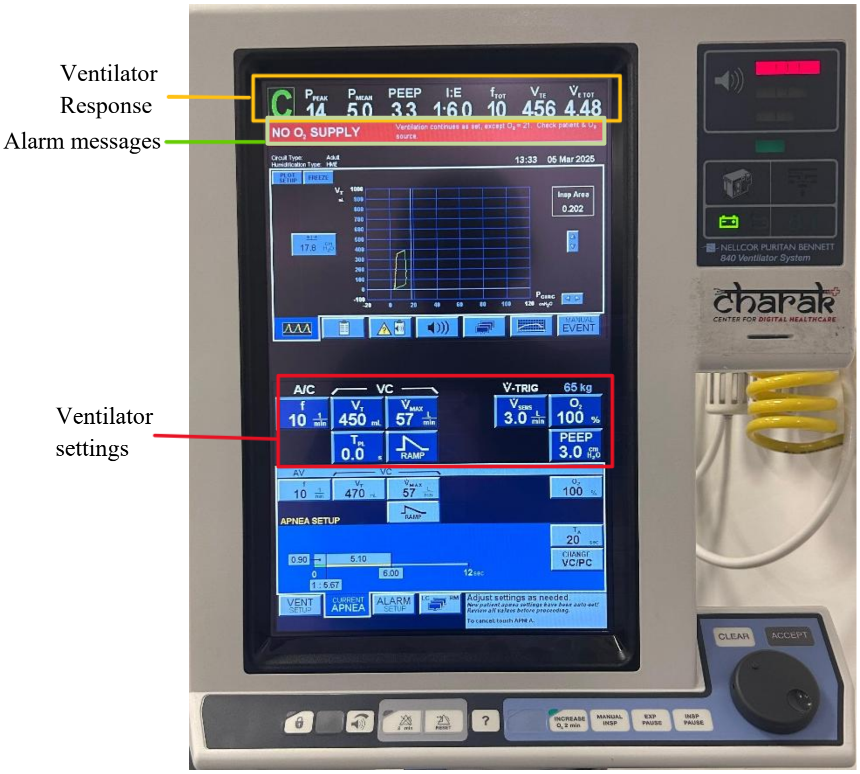


Figure 3 Puritan Bennett 840 ventilator screen

In this work, ventilator is a device that generates response parameters according to experiment parameters. It is connected to testchest (patient lung simulator) using the patient circuit and ET tube.

Basic Parameters of Mechanical Ventilation-

Tidal Volume (VT)- Measured in millilitres (ml), this parameter determines the amount of volume to be delivered during each machine breath.

Inspiratory Pressure (PI)- Measured in cmH2O, this parameter controls the maximum inspiratory pressure to be delivered to the patient during a pressure-controlled machine breath.

Respiratory Rate/Frequency (RR/f)- Measured in breaths per minute (BPM), this parameter determines the frequency for control breaths.

Fraction of Inspired Oxygen (FiO2)- Measured as a fraction of 100% oxygen, this parameter controls the oxygen concentration in the inspired gas.

Positive End-Expiratory Pressure (PEEP)- Measured in cmH2O, these determine the airway pressure above atmospheric pressure that exists at the end of expiration.

Inspiratory Time (TI)- Measured in seconds, this parameter controls the duration of the inspiratory phase of breath cycle.

Inspiratory and Expiratory Ratio (I: E Ratio)- This parameter controls the ratio of inspiration to expiration in relation to the machine rate.

Trigger (Trig. /Sens)- Can be measured in cmH2O or LPM, depending on the type of triggering system, this parameter determines the amount of inspiratory effort required by the patient before the ventilator will deliver an assisted breath, or demand flow in the case of a spontaneous breath.

Pressure Support (PS)- Measured in cmH2O, Pressure support provides a set amount of pressure during inspiration to support the spontaneously breathing patient.

Inspiratory Pause/ Plateau Time (T Plateau)- Measured in seconds, this parameter delays exhalation, therefore lengthening inspiration.

Sigh Frequency (Sigh f)- A sigh is a long and deep breath, define as a periodic deep breath 1.5 to 2.0 times the normal.

Continuous Positive Airway Pressure (CPAP)- Measured in cmH2O, these determine the constant level of pressure above atmospheric pressure is continuously applied to the upper airway.

Peak Flow (F/V')- Measured in litres per minute (LMP), this parameter controls the flow rate to be delivered to the patient during a machine breath.

Monitored Parameters of Mechanical Ventilation-

Peak Inspiratory Pressure (PIP)- Measured in cmH2O, Peak inspiratory pressure is the highest level of pressure applied to the lungs during inhalation. *Indicates the peak inspiratory pressure achieved during the last delivered breath.

Mean Airway Pressure (P Mean)- This indicates the mean (average) pressure in the airway over the last minute.

Exhale Tidal Volume (VTE)-This parameter provides the numerical representation of the patient's volume of exhaled air.

Exhale Minute Volume (MVE)- Represents the patient's exhaled tidal volume (mechanical and spontaneous) over time.

Minute Volume= Exhale tidal volume X Total Respiratory Rate

Total Respiratory Rate (RR TOT/ Ftot)- Represents the total breaths (machine and spontaneous) delivered by the ventilator during the last minute.

Plateau Pressure (P Plateau)- This parameter indicates the airway pressure during an inspiratory pause. This is used in the calculation of static compliance.

Resistance(R)- Measured In cmH2O/(l/s), Resistance describe the opposition to a gas flow entering the respiratory system during inspiration, which is caused by frictional forces.

R= ΔP/V'(Flow), Resistance is calculated as the ratio between the pressure driving a given flow and the resulting flow rate(V).

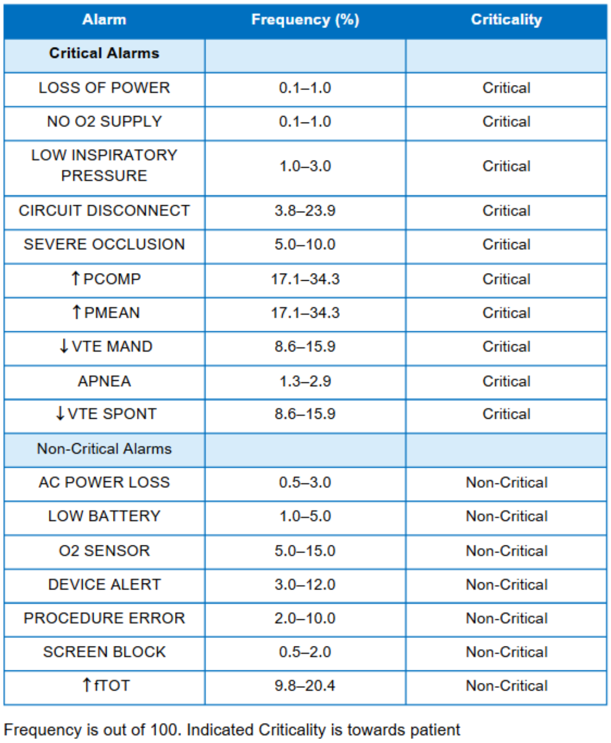
Compliance(C)- Measure in ml/cmH2O, Compliance describe the elastic property of the respiratory system including the lung and the chest wall.

C= ΔV/ΔP, Static compliance is the ratio in change in volume and the corresponding change in pressure.

Ventilator Alarms

Alarms are the visual and audible signals, intended for hospital staff, that indicate any abnormality with the patient connected to the device or fault in the device itself.

Ventilator alarm characteristics for the most common alarms:



Alarm	Frequency (%)	Criticality
Critical Alarms		
LOSS OF POWER	0.1–1.0	Critical
NO O2 SUPPLY	0.1–1.0	Critical
LOW INSPIRATORY PRESSURE	1.0–3.0	Critical
CIRCUIT DISCONNECT	3.8–23.9	Critical
SEVERE OCCLUSION	5.0–10.0	Critical
↑PCOMP	17.1–34.3	Critical
↑PMEAN	17.1–34.3	Critical
↓VTE MAND	8.6–15.9	Critical
APNEA	1.3–2.9	Critical
↓VTE SPONT	8.6–15.9	Critical
Non-Critical Alarms		
AC POWER LOSS	0.5–3.0	Non-Critical
LOW BATTERY	1.0–5.0	Non-Critical
O2 SENSOR	5.0–15.0	Non-Critical
DEVICE ALERT	3.0–12.0	Non-Critical
PROCEDURE ERROR	2.0–10.0	Non-Critical
SCREEN BLOCK	0.5–2.0	Non-Critical
↑fTOT	9.8–20.4	Non-Critical

Frequency is out of 100. Indicated Criticality is towards patient

Figure 4 Alarm frequency of most common ventilator alarms and corresponding criticality

Alarms can be broadly classified as patient and system alarms as shown in the figure below.

Patient alarm: Alarms that are caused due to patient condition.

System alarms: Alarms that are raised due to malfunction or damage to the equipment.

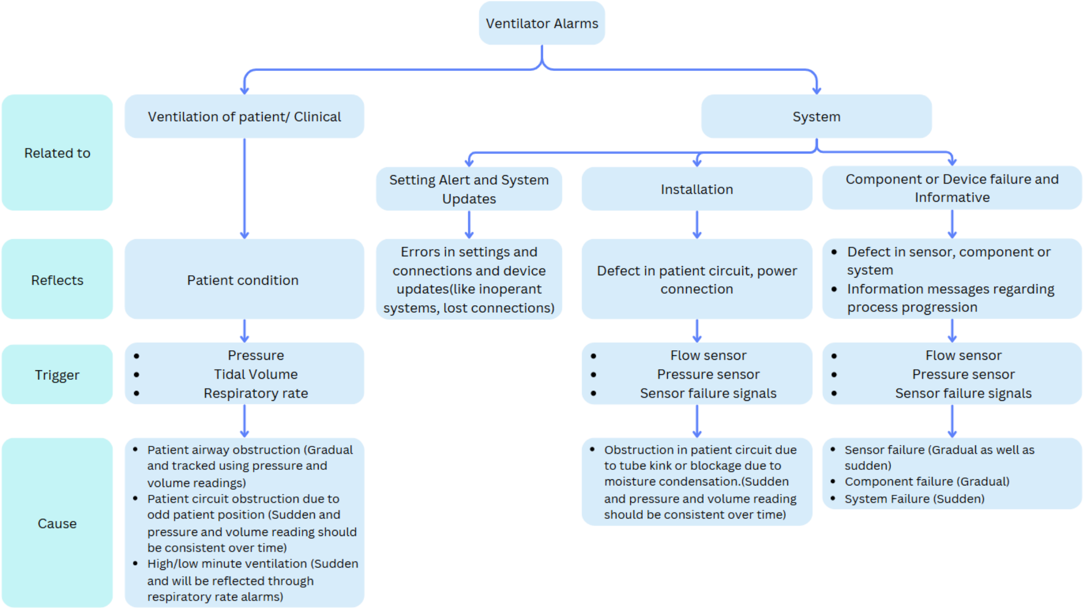

Ventilator Alarms
Related to
Ventilation of patient/ Clinical
System
Setting Alert and System Updates
Installation
Component or Device failure and Informative
Reflects
Patient condition
Errors in settings and connections and device updates(like inoperant systems, lost connections)
Defect in patient circuit, power connection
Defect in sensor, component or system
Information messages regarding process progression
Trigger
Pressure
Tidal Volume
Respiratory rate
Flow sensor
Pressure sensor
Sensor failure signals
Flow sensor
Pressure sensor
Sensor failure signals
Cause
Patient airway obstruction (Gradual and tracked using pressure and volume readings)
Patient circuit obstruction due to odd patient position (Sudden and pressure and volume reading should be consistent over time)
High/low minute ventilation (Sudden and will be reflected through respiratory rate alarms)
Obstruction in patient circuit due to tube kink or blockage due to moisture condensation.(Sudden and pressure and volume reading should be consistent over time)
Sensor failure (Gradual as well as sudden)
Component failure (Gradual)
System Failure (Sudden)

Figure 5 Alarm Classification on basis of origin of cause

This classification is done after studying the alarms of Puritan Bennett 840 [24]and Air liquid Taema Extend XT[25].

1.5.2. Para monitor/ Patient monitor

A **Para monitor**, more commonly known as a **multiparameter patient monitor**, is a clinical device that continuously measures and displays a patient's core vital signs—namely ECG (with heart rate), respiration rate, non-invasive blood pressure, blood oxygen saturation (SpO_2), pulse rate, and body temperature. Beyond these basics, most monitors offer plug-in modules for advanced metrics such as invasive blood pressure, end-tidal CO_2, respiratory mechanics, anesthetic gas concentrations, cardiac output (both invasive and non-invasive), and EEG bispectral index [26].

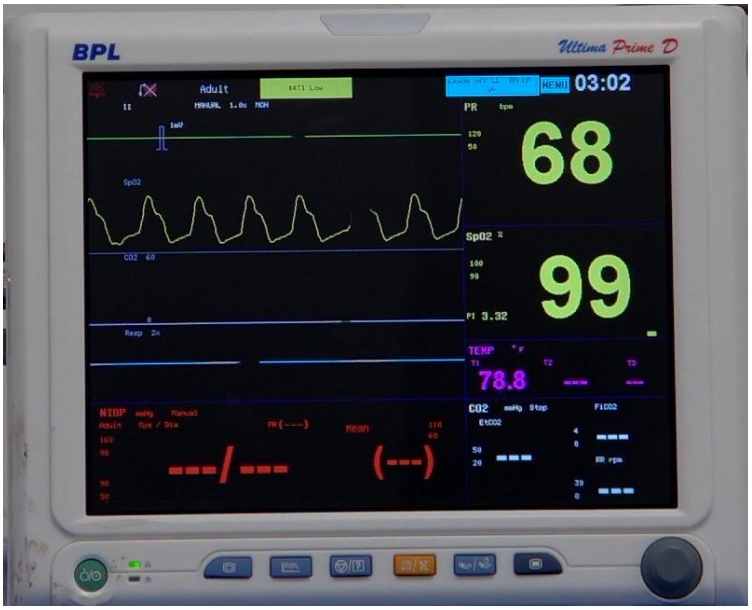


Figure 6 Para monitor/ Patient monitor

In routine use across emergency departments, operating rooms, ICUs, CCUs, and general wards, the patient monitor not only tracks these physiological parameters in real time but also compares them against preset thresholds. Configurable audible and visual alarms immediately alert clinical staff to any deviations, supporting prompt diagnosis and intervention. By providing trend data and event logging, these systems underpin critical decision-making and have been shown to reduce mortality among critically ill patients [26].

Structurally, modern monitors employ a modular "plug-in" architecture: each sensor or measurement function (e.g., blood oxygen probe, ECG leads, NIBP cuff) can be connected or removed as needed. Specialized modules—such as those for cardiopulmonary resuscitation quality monitoring—work in tandem with the main unit to evaluate and guide resuscitation efforts. Setup and operation involve a straightforward sequence of steps (powering on, electrode placement, cuff sizing, etc.), and careful attention to placement and hygiene helps ensure accurate readings and patient safety [26].

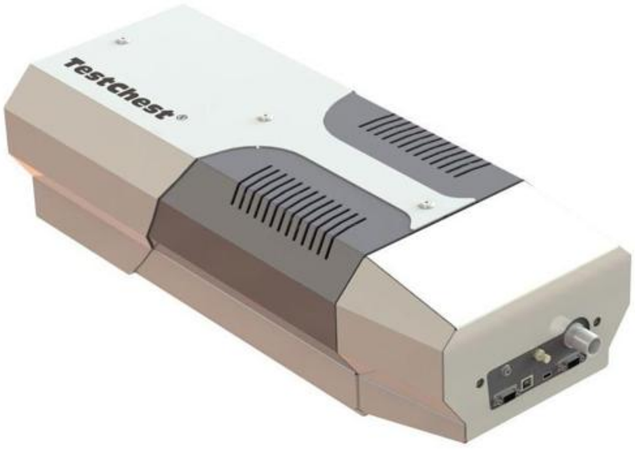
Designed for use in intensive care units, emergency departments, and operating rooms, it typically tracks parameters such as [26]:

- Electrocardiogram (ECG) and heart rate
- Non-invasive blood pressure (NIBP)
- Oxygen saturation (SpO_2) via pulse oximetry
- Respiratory rate
- Body temperature
- Optional modules (e.g., invasive blood pressure, end-tidal CO_2)

Para monitors are used to capture the required vitals through the data acquisition system in place. Further details about the use and setup are discussed in chapter 4.

1.6. Testchest

TestChest® V3 is essentially a tabletop model of the human heart and lungs, built for teaching and hands-on training. You can use it by itself as a standalone station or plug it into a larger patient-simulation system. Its main purpose is to work alongside a real mechanical ventilator and mimic how lungs inflate and deflate, how



Commercial Name:	TestChest
Model, Type:	V3
Serial No.:	00:50:C2:F1:10:00 - 00:50:C2:F1:1F:FF
Firmware Revision:	3_30_03_12 or higher
User Software:	TestChest^{LIFE}
Manufacturer:	Organis GmbH Tardisstrasse 225 7205 Zizers SWITZERLAND

Figure 7 Testchest V3

people breathe on their own, how oxygen and carbon dioxide move in and out of the blood, and even how changes in breathing affect blood oxygen saturation (SpO_2) and pulse pressure [27].

Software Control via TestChestLIFE [27]

The TestChestLIFE app breaks everything into five broad categories that you can adjust:

Lung Mechanics: Airway resistance (Raw) predicted functional residual capacity ($FRC_{pre}d$), overall respiratory system compliance (C_rs), plus more advanced settings like lower/upper inflection points on the compliance curve, percentage of lung collapsed at zero airway pressure, and time constants for recruitment/collapse.

Gas Exchange: Carbon dioxide production ($\dot{V}CO_2$), dead space volume (V_{daw}), oxygen diffusion limits (P_{diff}), and how quickly SpO_2 falls during an apnea.

Respiratory Control: The initial inspiratory effort ($P_{0.1}$), spontaneous breathing rate (f_{spont}), and even the shape of the spontaneous breathing waveform (apnea or one of fifteen predefined patterns).

Hemodynamic: Cardiac output (QT), heart rate, pulse pressure variation (POPv), and two parameters for heart–lung interaction (a time constant R_{Clh} and a phase delay T_{delay}), plus the amplitude of cardiogenic oscillations in airway pressure.

Special Effects: Leak level (none to large), a low-pass filter cutoff (to smooth displayed curves), and an FiO_2 override (if you want to force a specific oxygen fraction).

Behind the scenes, each of these settings feeds into internal algorithms that decide exactly how the bellows move, how the pressures change, and how quickly the lungs collapse or re-recruit. Those mechanical movements produce real pressure readings and tidal volumes that the ventilator “feels” just like a patient’s lungs. At the same time, TestChest® calculates SpO_2 and (eventually) end-tidal CO_2 based on gas exchange equations, then feeds those signals to the OxSim finger.

1.7. Simulated diseases

As the work revolves around ventilation, some most common respiratory conditions were simulated to generate relevant data. In this work, two diseases, Asthma and Ventilator Associated Pneumonia (VAP), were simulated and data for respective diseases was captured. Below subsections provide introduction and background information to how the parameters for simulation were selected and manipulated.

1.7.1. Asthma

Asthma is a chronic inflammatory disorder of the airways characterized by variable airflow obstruction and bronchial hyperresponsiveness. Clinically, it presents with wheezing, dyspnea, chest tightness, and cough, often worsening at night or early morning. Pathologically, asthma involves persistent airway inflammation driven predominantly by eosinophils, mast cells, and T helper 2 (Th2) lymphocytes, although non-Th2 (e.g., neutrophilic) phenotypes also occur. The underlying mechanisms include genetic predisposition (e.g., variants in the IL-4 receptor and ADAM33 genes), epigenetic modifications, and environmental factors such as allergen exposure (house dust mite, pollen), viral infections (rhinovirus), tobacco smoke, and occupational sensitizers [28], [29].

Multiple factors contribute to asthma onset and exacerbations. Atopy and a family history of allergic disease increase susceptibility by promoting IgE-mediated sensitization. Environmental allergens (e.g., aeroallergens, indoor mold) trigger Th2-driven inflammation, leading to airway eosinophilia and mast cell degranulation. Viral respiratory infections, especially in early childhood, can disrupt airway epithelial integrity and skew immune responses toward a pro-asthmatic phenotype. Tobacco smoke and air pollutants (ozone, particulate matter) enhance airway inflammation by generating oxidative stress and amplifying cytokine release (e.g., IL-5, IL-13). Obesity and psychosocial stress are emerging risk factors that may alter lung mechanics and immune regulation, further promoting an asthma phenotype [28], [29].

Physiologically, asthma is marked by several interrelated alterations:

1. **Airway Inflammation and Mucosal Edema:** Inflammatory mediators (histamine, leukotrienes, prostaglandins) increase vascular permeability, leading to plasma exudation into the airway submucosa. This mucosal edema narrows the airway lumen and contributes to airflow limitation [28], [29].

2. **Bronchial Smooth Muscle Constriction:** Hyperresponsive airway smooth muscle (ASM) contracts excessively in response to stimuli (e.g., methacholine, allergens), causing acute bronchoconstriction. This results in increased airway resistance and reduced forced expiratory volume in one second (FEV_1) [29].

3. **Mucus Hypersecretion:** Goblet cell hyperplasia and submucosal gland hypertrophy lead to overproduction of viscous mucus. Mucus plugs can occlude small airways, exacerbating airflow obstruction and promoting ventilation–perfusion mismatch [28].

4. **Airway Remodeling:** Chronic inflammation induces structural changes, including subepithelial fibrosis (thickening of the reticular basement membrane), ASM hypertrophy, and increased extracellular matrix deposition. Over time, these changes stiffen the airways and fix some degree of airflow limitation, making the disease less reversible [28], [29].

5. **Ventilation–Perfusion (V/Q) Mismatch:** Areas with bronchoconstriction and mucus plugging receive perfusion but limited ventilation, producing right-to-left shunts and hypoxemia. Conversely, overventilated but underperfused regions contribute to dead space, further impairing gas exchange [29].

6. **Dynamic Hyperinflation:** During exacerbations, expiratory flow limitation prevents complete lung emptying. This leads to air trapping and an increase in end-expiratory lung volume (auto-PEEP), which increases the work of breathing and may cause respiratory muscle fatigue [28].

In summary, asthma arises from genetic and environmental interactions that drive chronic airway inflammation. The key pathophysiological features include episodic bronchoconstriction, mucus hypersecretion, and structural remodeling, all of which culminate in varying degrees of airflow obstruction, V/Q mismatch, and increased work of breathing. Appropriate management targets both inflammatory pathways (e.g., inhaled corticosteroids) and bronchoconstriction (e.g., short-acting β_2-agonists) to mitigate symptoms and prevent progressive airway remodeling.

1.7.2. Ventilator Associated Pneumonia (VAP)

Ventilator-associated pneumonia (VAP) is defined as pneumonia that develops at least 48 hours after endotracheal intubation and initiation of invasive mechanical ventilation, in a patient whose respiratory tract was not infected at the time of intubation. Early-onset VAP (post-intubation days 1–4) typically involves antibiotic-sensitive pathogens, whereas late-onset VAP (day 5 onward) is more often due to multidrug-resistant organisms [30], [31], [32], [33], [34].

VAP arises because invasive ventilation breaches normal airway defenses and promotes microaspiration of colonized secretions. Specifically, the endotracheal tube bypasses the glottic barrier and inhibits the cough reflex, allowing oropharyngeal flora to pool above the cuff and leak into lower airways [30], [31]. Sedation, supine positioning, and neuromuscular blockade further impair mucociliary clearance and cough, facilitating aspiration [30], [31], [34]. Within days of ICU admission—especially under broad-spectrum antibiotic pressure—the oropharynx and trachea become colonized by nosocomial pathogens (e.g., *Acinetobacter baumannii*, *Pseudomonas aeruginosa*, *Staphylococcus aureus*) [31], [33]. Once these bacteria reach the distal airspaces, alveolar macrophages and neutrophils mount an inflammatory response, leading to alveolar exudation and consolidation [30], [31], [34]. Biofilm formation on the endotracheal tube further serves as a reservoir for pathogens, which can be dislodged during suctioning and inoculate the lung [31]. Key risk factors include mechanical ventilation duration > 48 hours; supine positioning; sedation and paralysis; reintubation; nasogastric tubes; and prior antibiotic use, which selects for resistant organisms [30], [31], [33].

By consensus, VAP manifests no earlier than 48 hours after endotracheal intubation. Early-onset VAP (days 1–4 of ventilation) generally involves community-type organisms (e.g., MSSA, *Haemophilus influenzae*) and carries a more favorable prognosis, whereas late-onset VAP (day 5 or later) is associated with nosocomial, multidrug-resistant pathogens (e.g., *A. baumannii*, *P. aeruginosa*), resulting in higher morbidity and mortality [30], [33]. The highest risk of developing VAP occurs within the first 10 days of mechanical ventilation; thereafter, although the daily risk diminishes, it persists as long as the endotracheal tube remains in place [31].

Physiological Changes in VAP:

When VAP develops, the following pathophysiological alterations occur:

1. **Alveolar Inflammation and Consolidation:**

   Bacterial pathogens reaching the alveoli trigger innate immune activation, leading to neutrophil recruitment, release of inflammatory mediators, and alveolar-capillary membrane injury. This results in consolidation and exudation, impairing gas exchange [30], [31].

2. **Ventilation–Perfusion (V/Q) Mismatch:**

   Consolidated or fluid-filled alveoli receive perfusion but no ventilation (right-to-left shunt), causing refractory hypoxemia; concomitantly, airway obstruction and bronchospasm elevate physiologic dead space, leading to CO_2 retention [31].

3. **Impaired Oxygenation:**

   As V/Q mismatch worsens, the PaO_2/FiO_2 ratio decreases. Clinicians often escalate FiO_2 and positive end-expiratory pressure (PEEP) to maintain adequate oxygenation, which may exacerbate lung injury [30], [31].

4. **Systemic Inflammatory Response (SIRS):**

   Local cytokine release (e.g., IL-1, TNF-α) propagates a systemic response characterized by fever, tachycardia, leukocytosis, and—in severe cases—

capillary leak and hypotension. If bacteremia ensues, patients may progress to septic shock [31], [32], [34].

5. **Altered Respiratory Mechanics:**

   Consolidation reduces lung compliance, increasing plateau pressures required to deliver target tidal volumes; simultaneously, airway inflammation and secretions increase airway resistance, raising peak inspiratory pressures [31].

6. **Hemodynamic Consequences:**

   Hypoxic pulmonary vasoconstriction in consolidated lung regions elevates pulmonary vascular resistance, potentially overloading the right ventricle. Systemic vasodilation and capillary leak from sepsis can reduce preload and cardiac output, compounding tissue hypoxia [31].

7. **Risk of Progression to ARDS:**

   Unchecked alveolar-capillary damage may evolve into diffuse alveolar damage (DAD) and acute respiratory distress syndrome (ARDS), characterized by hyaline membrane formation and severe, refractory hypoxemia [31].

8. **Multiorgan Dysfunction:**

   Severe VAP—especially when caused by multidrug-resistant Gram-negative organisms—can precipitate acute kidney injury, coagulopathy, and multiorgan failure, markedly increasing mortality risk [32], [33].

This concludes the introduction to the thesis. The introduction encapsulates the whole idea of digital transformation in healthcare sector with introduction to all the technologies, medical equipment used and simulated medical conditions.

1.8. Organization of thesis

Chapter one introduces Industry 4.0, Digital healthcare and all the technologies, equipment, and medical conditions that are utilized in this work.

Chapter Two will be devoted to an exhaustive review of the literature, critically evaluating and integrating previous studies on the subject, pinpointing gaps in the existing knowledge, articulating the research problem and objectives, and exploring the pertinent theories and frameworks.

Chapter three discusses the current ICU decision chain and proposes a solution to optimize it.

Chapter four elaborates on the methodology used to build the proposed solution. It talks about the data involved, the experimentation performed and the data generation and acquisition method, mapping of alarm, causes and responsive actions, and finally the criticality calculation method.

Chapter five and six deals with results and discussions, and conclusions that are the outcomes of the thesis work. Finally, chapter seven emphasizes the future potential and scope of this work.

Chapter 2: Problem statement formulation

2.1. Literature review

The Intensive Care Unit (ICU) is a hospital setting designed to manage patients with life-threatening conditions, necessitating continuous monitoring and rapid interventions. ICUs handle complex cases—severe sepsis, multiorgan failure, acute respiratory distress—and employ advanced life-support technologies (ventilators, infusion pumps, renal replacement therapy) under multidisciplinary teams of intensivists, critical-care nurses, respiratory therapists, and pharmacists [35], [36]. Although life-saving, this high-acuity environment is prone to unique challenges, particularly related to resource constraints, diagnostic complexity, patient transitions, and the cumulative burden of alarms. This review examines general ICU issues, defines "alarm fatigue" across seminal studies, highlights alarm-specific problems, and summarizes proposed solutions in chronological order.

Problems faced in an ICU

ICUs frequently operate under resource and infrastructure limitations, especially in low- and middle-income regions, where shortages of trained staff and disrupted supply chains hinder standardized care delivery [35]. High patient acuity and diagnostic complexity compound these challenges: conditions like septic shock and acute respiratory distress syndrome (ARDS) demand prompt recognition, yet heterogeneous presentations delay diagnosis [37]. Staffing pressures such as chronic nursing shortages, low nurse-to-patient ratios elevate workloads, increasing the likelihood of medical errors, infection transmission, and adverse events [36]. Diagnostic errors also arise from frequent interruptions, cognitive overload, and inconsistent adherence to evidence-based protocols (e.g., variations in "Wake Up and Breathe" compliance) [38]. Patient transitions—from ICU to step-down units can produce care-continuity gaps, communication failures, and mixed emotional responses among patients, families, and providers, further complicating recovery [39]. Meanwhile, survivors often endure post–intensive care syndrome (PICS) long-term physical weakness, cognitive deficits, and psychiatric sequelae stemming from prolonged immobility, extended sedation, and sleep disruptions during their ICU stay

[37]. Finally, environmental factors such as high ambient noise (average 68–75 dBA) and suboptimal unit design hinder both patient rest and staff performance [40].

Alarm Fatigue definition

Deb and Claudio (2015) define alarm fatigue as a combined state of heightened mental workload and negative affect (boredom, apathy, distrust) resulting from excessive false or nonactionable alarms; measured via NASA-TLX and affect scales during direct observations of ICU nurses and unit clerks [41]. Walsh and Waugh (2020) describe alarm fatigue implicitly as clinician desensitization caused by fewer than 15 % of mechanical ventilation (MV) alarms being clinically relevant, leading to sensor overload and delayed responses [42]. Dills (2017) frames alarm fatigue operationally around middleware-filtered scenarios: RTs exposed to > 19 000 ventilator alarms/day become unable to differentiate actionable events amid a flood of nonactionable high inspiratory pressure or high respiratory rate alarms [43]. The AAMI/ACCE (2006) white paper characterizes alarm fatigue as clinicians becoming "cognitively numb" to frequent false alarms (85–99 % false positive), leading to alarm disabling or ignoring and occasional failure to recognize true emergencies [44]. Stokes, Manzoor, and Cvach (2017) imply alarm fatigue through RTs reporting frequent nuisance ventilator alarms, undermining confidence in alarm reliability and prompting avoidance behaviors in alarm customization [38].

Alarm fatigue underlies several interrelated ICU problems. High false or nonactionable alarm rates—often 85–90 % of MV alarms and 88.8 % of arrhythmia alarms—overwhelm clinicians, reducing trust and elongating response times [40], [42]. Ambient noise levels exceeding 82 dBA from alarms further desensitize staff and mask true alerts [36], [40]. Inconsistent alarm terminology and priority coding across ventilator brands cause confusion: for instance, 89.8 % of Hamilton G5 alarms are "high priority" versus only 8.6 % of Puritan Bennett 840 alarms in the same ICU, despite similar physiologic triggers [43]. Alarm limit settings rarely track actual patient parameters—upper respiratory rate limits change by only 1 breath/min for a 10 breaths/min change in measured rate—resulting in either nuisance alarms or delayed detection of critical events [43]. Furthermore, redundant alarm notification cascades (e.g., central station + nurse-call + middleware to Wi-Fi phones) can amplify a single event into multiple alerts, with each ventilator averaging 7 initial alarms/hour and 2.8 secondary notifications/hour, equating to ≈ 9.8 notifications/ventilator-hour (one every ≈ 6 minutes) [43].

Industry-level initiatives began in 2006 when the ACCE Healthcare Technology Foundation recommended adopting IEC 60601-1-8 to standardize alarm tones and priorities, alongside hospital-wide policies that specify which alarms require immediate intervention [44]. They also advised appointing "alarm champions," performing regular alarm audits, and embedding alarm-system training into staff orientation and annual competencies. These measures aimed to reduce nuisance alarms, harmonize clinician interpretation, and reinforce alarm-management accountability.

Building on that foundation, Yang et al. (2012) developed a prototype intelligent ventilator alarm system using a PIC32MX microcontroller [38]. By continuously sampling ventilator parameters (P_{peak}, R_{exp}, F_{exp}, end-PEEP) via RS-232 and applying context-specific logic (e.g., $P_{peak}\uparrow + R_{exp}\uparrow + F_{exp}\downarrow \rightarrow$ "suction needed"), the system generated targeted alerts that were wirelessly transmitted via ZigBee to a nurse-station interface. Bench testing of sputum impaction and airway leak scenarios demonstrated significant reductions in false alarms compared to traditional high-pressure thresholds.

In 2015, Deb and Claudio quantified alarm fatigue in ICU staff through NASA-TLX (mental workload) and affect scales [41]. Their observational study linked high alarm frequency, environmental noise, and staffing ratios to elevated workload and negative affect, identifying personality traits as moderating factors. They recommended optimizing nurse-to-patient ratios, tailoring training to individual profiles, and implementing noise-reduction strategies rather than focusing solely on alarm volume reduction.

From 2017 to 2018, multiple institutions implemented middleware and quality-improvement projects. Dills (2017) deployed middleware at Hospital for Special

Care to filter ventilator alarms: patient-specific HIP and HRR alarms that self-resolved within 90 s were suppressed, while actionable alarms (disconnect, low minute ventilation) were forwarded to pagers and Wi-Fi phones, markedly reducing nonactionable alerts [43]. De Vaux et al. (2017) formed a multidisciplinary alarm management team at Yale New Haven Hospital, using the AACN toolkit to identify PVC alarms as the main nuisance; interventions—staff education, zone-based response, disabling PVC defaults, enabling continuous QTc monitoring—achieved a 77 % reduction in audible alarms and raised alarm customization from 39 % to 87.5 % with no adverse events [36]. Stokes et al. (2017) at Johns Hopkins simplified ventilator alarm guidelines and provided targeted RT education, increasing alarm customization from 27 % to 40 % within 24 h of ventilator initiation by addressing overly complex policies and notification gaps [38]. Meanwhile, Villanueva et al. (2018) implemented staff education, "quiet hours," and "Quiet, Please" signage to reduce ICU noise from 68 ± 5 dBA to 60 ± 4 dBA ($p < 0.001$), improving patient sleep scores and indirectly mitigating alarm fatigue [40].

In 2019 and 2020, researchers synthesized insights and refined best practices. Scott et al. (2019) conducted a PRISMA-guided review showing ICU patients face 150–190 alarms per day (MV alarms = 11.7–42.2 %, 82–83 dBA), linking generic thresholds to nuisance alerts [45]. They advocated "smart alarms" (composite triggers, adaptive delays), middleware for off-site notifications, standardized MV alarm protocols, and interdisciplinary education to restore trust. Walsh and Waugh (2020) echoed these recommendations, emphasizing individualized thresholds based on patient baselines, intelligent alarm features (paired triggers, escalation hierarchies), and biomedical device integration to route alarms through tiered dashboards [42]. Cvach et al. (2020) performed a prospective study using Capsule Axon bridges on PB840 and G5 ventilators, revealing 7 ± 4 alarms/hour with 40 % persisting > 15 s (triggering 2.8 ± 1.8 secondary notifications, ~ 9.8 cascades/hour). They highlighted manufacturer priority discrepancies and poor alarm-limit/parameter correlations, proposing continuous logging to guide patient-specific adjustments and unified priority definitions [43]. Lin et al. (2020) applied HFMEA to intrahospital transport, introducing mnemonic "reminder-assisted briefings"

(VITAL, STOP) that reduced IHT-related adverse events from 1.08 % to 0.23 % ($p = 0.01$) and increased task completeness from 80.8 % to 96.5 % ($p < 0.001$) [37]. Recent work has advanced both research agendas and technological solutions. Scott (2021) issued five research priorities: MV alarm reliability, patient-specific thresholds, off-bed notification strategies, clinician competency frameworks, and analyses of invasive versus noninvasive mode alarms [46]. Coldewey et al. (2021) systematically reviewed ventilation-device usability, identifying 51 failure modes—ambiguous labels, hidden power switches, deep menu hierarchies, inconsistent terminology (e.g., "Tplat" vs. "Tpause"), and confusing alarm color coding—and recommended ISO 19223 for uniform labeling, IEC 60601-1-8 for standard tones, UI improvements, and consistent alarm coding across brands [47]. Asadi et al. (2022) surveyed 140 COVID-19 ICU nurses using a 13-item alarm fatigue questionnaire (range 8–44) and a 24-item moral distress scale (range 0–96), finding moderate alarm fatigue (19.08 ± 6.26) inversely correlated with ventilator/alarm training ($r = -0.25$, $p < 0.01$) and higher fatigue among female/PhD-level nurses; they recommended regular hands-on training and minimizing rotating shifts [48]. Li and Ge (2021) proposed a representation-learning framework that constructs dynamic knowledge graphs and Probabilistic State Machines to compute real-time imminence scores, suppressing low-risk alarms without predefined patterns and matching LSTM accuracy at ≈ 100× throughput [39], [40]. Finally, Li et al. (2024) described a Node.js- and WebSocket-based web monitoring system integrating RS-232 microcontrollers, PostgreSQL storage, and React Native mobile clients to filter noncritical alarms; in a 50-patient pilot, they achieved a 30 % reduction in alarm-response latency (12 ± 3 s vs. 17 ± 5 s; $p < 0.01$) and high usability scores (≥ 4.2/5) [49].

2.3. Research Gap

Despite extensive documentation of alarm-related hazards in ICUs, false and nonactionable alarms persist as a significant safety concern. Early observational studies quantified how nurses and respiratory therapists are exposed to hundreds of alarms per shift, leading to delayed response times and, in some instances, missed critical events [41], [43]. Even with guidelines advising customization of alarm thresholds, alarm volumes often rebound when new devices are introduced or when defaults are reset. In practice, alarm overload continues to erode clinician trust, illustrating that awareness alone is insufficient to curb the problem.

Many publications advocate policy frameworks and staff education as key remedies. The ACCE Healthcare Technology Foundation (2006) recommended standardizing alarm tones (IEC 60601-1-8), appointing "alarm champions," and conducting regular audits [44]. Similarly, De Vaux et al. (2017) reported that multidisciplinary teams and PDSA cycles reduced specific nuisance alarms by 77 % through staff training and default setting adjustments [36]. Stokes et al. (2017) showed that simplifying ventilator alarm guidelines and focused RT education increased alarm customization from 27 % to 40 % [38]. However, these interventions often rely on ongoing compliance and repeated training, which can wane under high patient loads or frequent staff turnover.

To complement policies and training, several groups have developed technological tools. Yang et al. (2012) created a bench-tested intelligent alarm that filters artifacts by analysing ventilator parameters in real time [38]. Dills (2017) implemented middleware to suppress short-lived ventilator alarms, forwarding only actionable events [43]. Later, Walsh and Waugh (2020) and Li & Ge (2021) proposed smart-alarm features pairing related alarm cues, adaptive delays, and PSM-based imminence scoring to further reduce false positives [40], [42]. Li et al. (2024) demonstrated a web-based monitoring system that lowered alarm-response latency by 30 % in a 50-patient pilot [49]. Despite promising results, these solutions have seen limited real-world adoption due to scalability challenges, compatibility issues with diverse device inventories, and the need for substantial IT infrastructure.

In summary, existing efforts—whether guideline-driven, educational, or technological—tend to address individual aspects of alarm fatigue without delivering a fully integrated solution. Guidelines and training improve staff behaviour only while actively reinforced; intelligent alarms and middleware reduce noise but often require bespoke setups that are difficult to generalize across ICUs. A research gap remains for a cohesive framework that combines adaptive, patient-specific thresholds; universally consistent user interfaces; seamless integration with electronic health records; and minimal additional burden on frontline staff. Addressing this gap is essential to transform isolated successes into sustained, ICU-wide improvements.

So, this project focusses on fabricating a smart decision support system to analyse the criticality of each alarm based on patient condition as well as identify cause form ventilator parameter fluctuations and suggest responsive actions to the staff.

2.4. Research objectives

- To detect actual cause of the fluctuations in the equipment readings
- To classify alarms as Patient and System alarms
- To find out the criticality of the alarms
- To recommend actions appropriate for raised alarm

Chapter 3: Proposed Solution

3.1. Current ICU Decision Chain

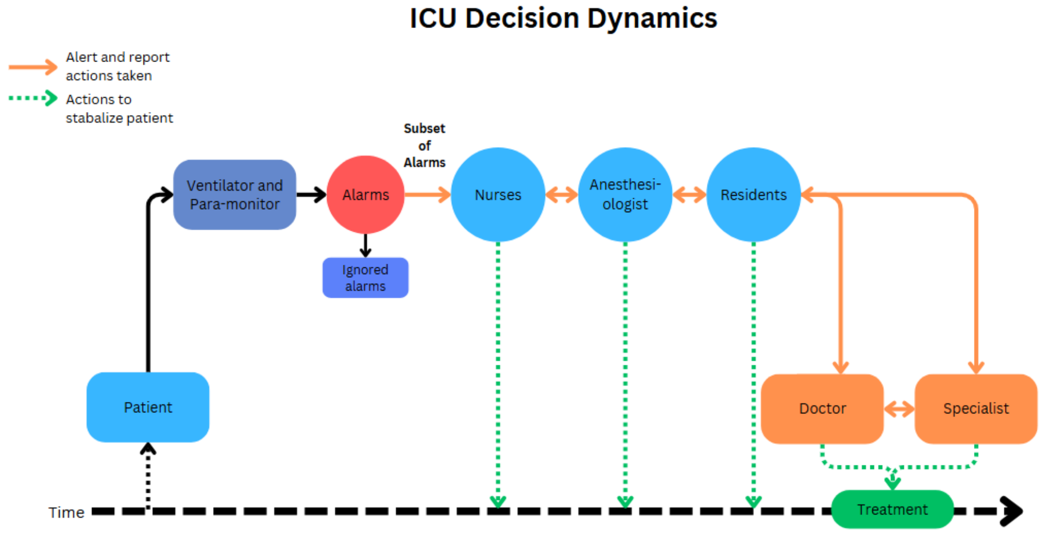


Figure 8Current ICU Decision Dynamics

Just like any workplace, an ICU has an order of authority and task compartmentalization. The above diagram depicts the decision-making hierarchy in an ICU. A patient in ICU is always hooked to a device called Para monitor. This device is used to monitor vital signs of a patient, to get an alert when the vitals deviate out of acceptable ranges. Similarly, ventilators are also a device that is used more frequently in an ICU. Whenever there is abnormality with the patient, these devices when hooked to a patient will generate alerts or alarms.

These alarms can be because of various reasons, so the actions performed for each one will be different for different circumstances. Some alarms may be ignored, some may be fixed with minor adjustments, some may need clinical intervention, and others may be due to faulty equipment. There are numerous possibilities.

Nurses are mostly the first individuals to be alerted by any change in patient condition, as they are stationed near the patient all the time. They have a particular set of actions and check list to perform and check to stabilize the situation. Anything beyond their scope needs to be alerted to Resident Doctors and Anesthesiologists (in case of mechanical ventilation). Any decision regarding equipment settings and clinical intervention is taken by these roles. Decisions regarding involving a specialist as the situation demands is also a decision a doctor makes. Final treatment course decisions are taken by Senior doctors and specialists after looking at the report from their sub-ordinates.

Everyone has some time delay in their actions depending on how the information was passed and what actions were taken by the immediate sub-ordinate. This is indicated using green dotted line.

When an alarm occurs, the first three healthcare professionals should stabilize the situation as soon as possible as delayed action may endanger the patient's life. Mostly, senior doctors are informed about patient progress twice a day, which is a large timespan for some underlying issues to go unresolved.

The proposed solution targets these issues.

3.2. Where does proposed solution fit in?

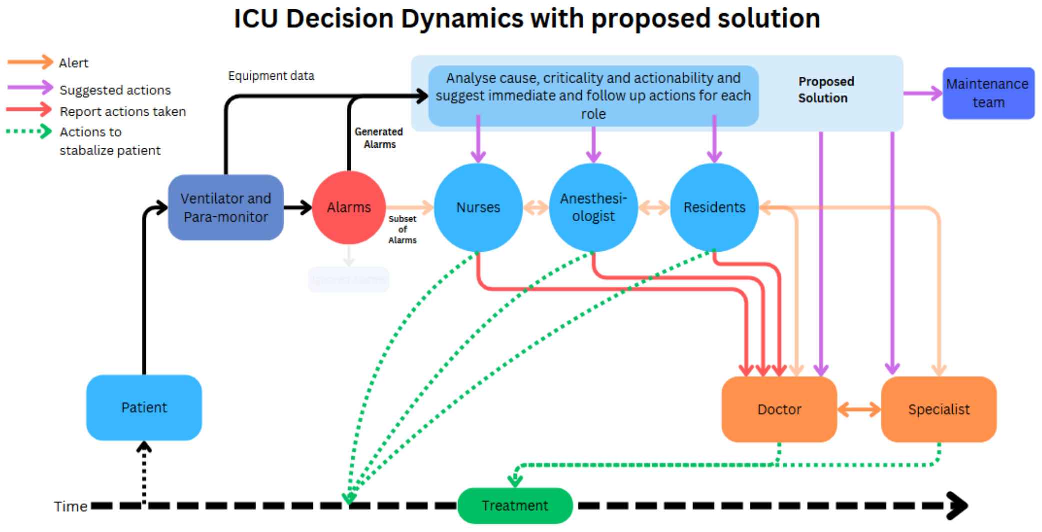


Figure 9 ICU Decision Dynamics with proposed solution

The proposed solution aims to resolve this issue by firstly determining critical or actionable alarms and suggesting the most appropriate actions.

This objective is achieved by capturing real time data from equipment, analyzing it to derive the cause of alarm and suggesting actions based on type of cause. This solution uses Industry 4.0 technologies, meaning digitalization and real time analysis, which reduces the time delay in the actions of medical professionals.

The purple arrows in the diagram represent the information co-ordination between solutions and the medical professionals. Also, the actions taken by primary attenders will have no delay as the process can be parallel, instead of being in series like that in conventional systems.

Benefits of the solution:

- Reduction in response time
- Allows actions by multiple staff members to be performed simultaneously
- Senior Doctors can be alerted at any instance of emergency with a full report
- Reduction in time needed to adjust treatment plans

Chapter 4: Methodology

This chapter talks about the approach used to achieve the set objectives and steps followed to decipher the issue and suggest actions to the staff on ground.

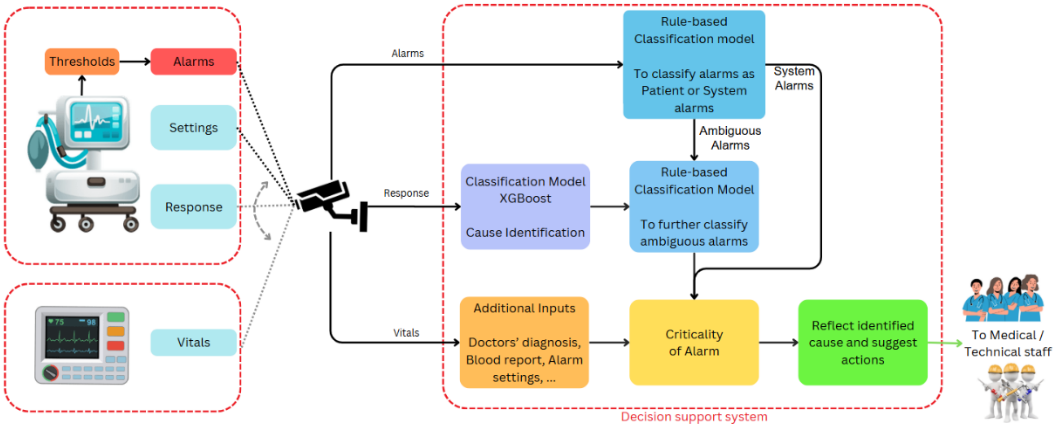

Thresholds
Alarms
Settings
Response
Vitals
Alarms
Response
Vitals
Rule-based Classification model
To classify alarms as Patient or System alarms
System Alarms
Ambiguous Alarms
Classification Model XGBoost
Cause Identification
Rule-based Classification Model
To further classify ambiguous alarms
Additional Inputs
Doctors' diagnosis, Blood report, Alarm settings, ...
Criticality of Alarm
Reflect identified cause and suggest actions
To Medical / Technical staff
Decision support system

Figure 10 Working of proposed solution

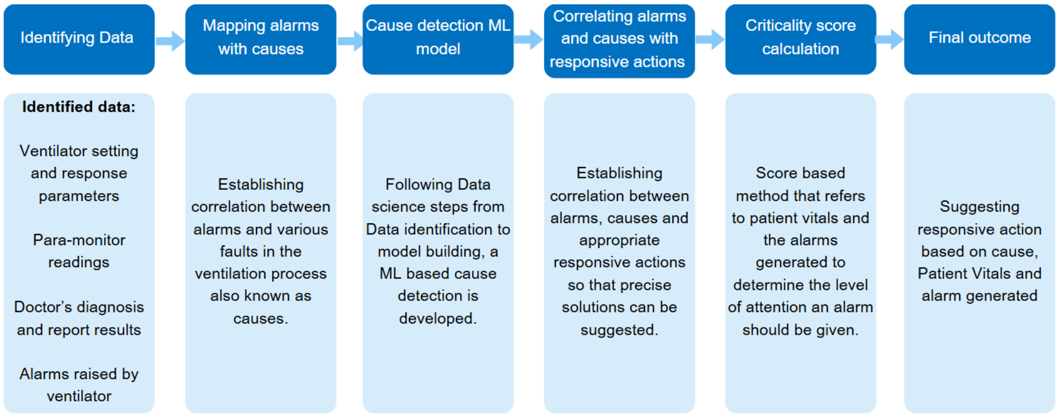
The diagram represents the complete flow of information, algorithms and backend analytics performed to get to a solution. Starting with the alarm generated, it is triggered when certain parameters cross the set threshold or the device needs maintenance. These alarms are to be labelled as critical or non-critical.

The first step is to classify them as patient alarms and System alarms. A simple rule-based classifier sorts of patient and system alarms. There is a set of alarms that can reflect both patient conditions as well as system failures/faults. So, these alarms are labelled Ambiguous alarms, as they cannot be told apart at this stage.

Simultaneously, Ventilator parameters are being analyzed for cause detection. This data driven process detects any abnormalities in data and tries to distinguish the type of failure or fault in the system or at the patient end. This is used as the most probable cause for the alarms. Correlating alarms and causes helps further classification of ambiguous alarms as patient or system alarms.

Once the alarm is classified as patient or system alarm, this information is used to calculate the criticality of that alarm. Along with alarm type, other major contributions to determining the criticality of any alarm are the patient vital conditions at that given instance and the Artillery blood gases (ABG) report parameters. The score calculated tells whether the alarm is critical or not, in other words, it tells us if alarm should be attended or ignored.

The results are displayed to the hospital staff, which contains the cause of the alarm and some responsive actions that each staff member can perform to resolve the issue or stabilize the patient.


Identifying Data
Mapping alarms with causes
Cause detection ML model
Correlating alarms and causes with responsive actions
Criticality score calculation
Final outcome

Identified data:

Ventilator setting and response parameters

Para-monitor readings

Doctor's diagnosis and report results

Alarms raised by ventilator

Establishing correlation between alarms and various faults in the ventilation process also known as causes.

Following Data science steps from Data identification to model building, a ML based cause detection is developed.

Establishing correlation between alarms, causes and appropriate responsive actions so that precise solutions can be suggested.

Score based method that refers to patient vitals and the alarms generated to determine the level of attention an alarm should be given.

Suggesting responsive action based on cause, Patient Vitals and alarm generated

Figure 11 Methodology

The steps taken to achieve the outcome are as follows:

- Data
- Mapping alarms with causes
- Cause detection ML Model
- Correlating alarms and causes with responsive actions
- Criticality score calculation

These steps are elaborated ahead.

4.1. Identifying Data

"Data" is defined as a representation of facts, concepts, or instructions in a formalized manner, suitable for communication, interpretation, or processing by humans or by automatic means[50]. Data can be measured, collected, reported, and analyzed, whereupon it is often visualized using graphs, images, or other analysis tools. Raw data ("unprocessed data") may be a collection of numbers or characters before it's been "cleaned" and corrected by researchers. It must be corrected so that we can remove outliers, instruments, or data entry errors. Data processing commonly occurs in stages, and therefore the "processed data" from one stage could also be considered the "raw data" of subsequent stages. Field data is data that's collected in an uncontrolled "in situ" environment. Experimental data is the data that is generated within the observation of scientific investigations. Data can be generated by:

- Humans
- Machines
- Human-Machine combines.

It can often generate anywhere where any information is generated and stored in structured or unstructured formats

The variables that are identified to understand the alarms and the patient conditions are: Alarm messages from ventilator, Ventilator response parameters, Ventilator settings, para monitor responses lab reports and doctors' diagnosis.

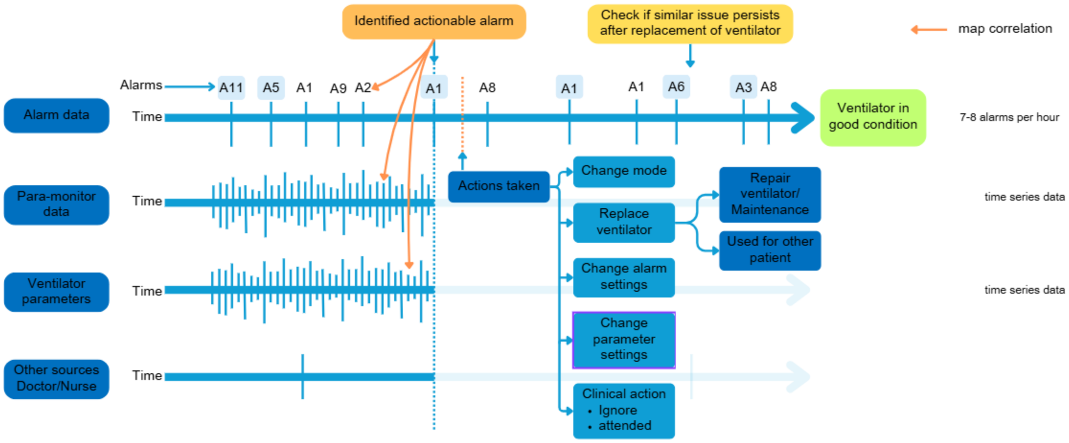


Figure 12 Required data characteristics

Definitions:

Ventilator Settings and Alarm Settings: These parameters are set on a ventilator before attaching the patient so that the patient is stabilized. The values or range of values are decided by the Anesthesiologist according to patient condition.

Ventilator Alarms: It is a message representing any abnormality with the patient or the system (ventilator), generated by the ventilator, to alert the hospital staff. It occurs on the ventilator screen with associated remedies.

Para Monitor Response: Parameters that depict the patient vitals in real time.

Ventilator Response Parameters: These parameters show the patient's response to the mechanical ventilation provided by the ventilator.

Arterial Blood Gasses (ABG) Report: This report provides information about the important factors of blood that are directly related to ventilation or respiratory system functioning.

The table below shows the data generation frequency of each group of parameters. Each parameter in a group has the same frequency as that of the group of parameters.

Table 1 Generation frequency of each data group

Data	Source	Generation Frequency
Ventilator Settings	Ventilator	continuous
Ventilator Alarm Settings	Ventilator	once every setup
Ventilator alarms	Ventilator	7-8 per hour on average
Para monitor Readings	Para monitor	continuous
Ventilator Response	Ventilator	continuous
ABG report	Lab results	Once or per day

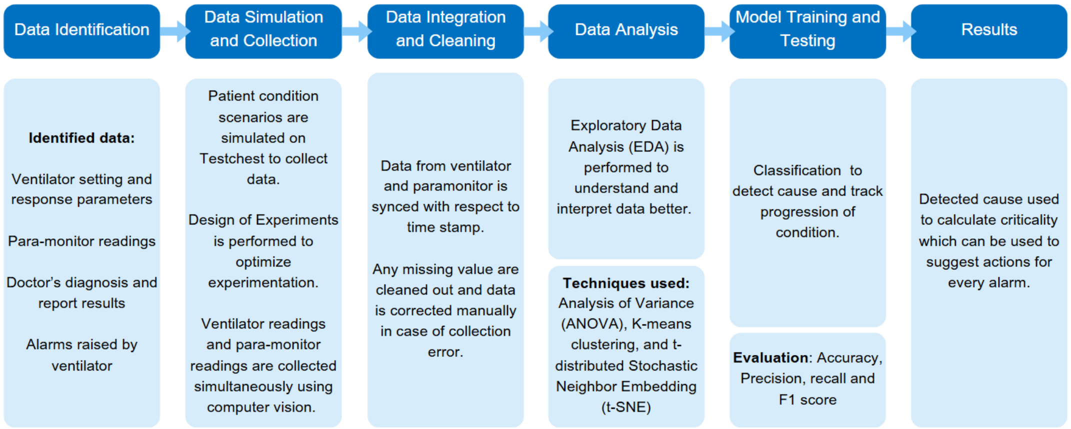
Details of each parameter are given below[24]:

Table 2 Name, representation and unit of every parameter

Parameters	Representations	Units
Ventilator Settings		
Ventilation Mode	-	-
Inspiratory Pressure	PI	cm H2O
Pressure support	PS	cm H2O
Positive End Expiratory Pressure	PEEP	cm H2O
Tidal volume	Vt	ml
Maximum Minute ventilation	Vmin	L/min
Respiratory Rate	F	bpm- breaths per minute
Inhalation time to Exhalation time ratio	I:E	-
Fraction of inspired oxygen	FiO2	%
Ventilator Alarm Settings		
Peak pressure	Ppeak	cm H2O
Respiratory Rate	Ftot	bpm- breaths per minute
Minute ventilation	Vetot	L/min
Tidal volume mandatory	Vtmand	ml
Tidal volume spontaneous	Vtspont	ml
Response Parameters		
Ventilator alarms		
Alarm message	-	-
Para-monitor Readings		
Heart Rate	HR/ PR	bpm- beats per minute
Oxygen Saturation	SpO2	%
Blood Pressure	BP	mm of Hg
Temperature	Temp	F
Ventilator Response		
Type of ventilation	-	-
Peak Pressure	Ppeak	cm H2O
Positive End Expiratory Pressure	PEEP	cm H2O
Tidal volume	Vt	ml
Minute ventilation	Vetot	L/min

Respiratory Rate	Ftot	bpm- breaths per minute
Inhalation time to Exhalation time ratio	I:E	-

4.2. Cause detection



Data Identification

Identified data:

Ventilator setting and response parameters

Para-monitor readings

Doctor's diagnosis and report results

Alarms raised by ventilator

Data Simulation and Collection

Patient condition scenarios are simulated on Testchest to collect data.

Design of Experiments is performed to optimize experimentation.

Ventilator readings and para-monitor readings are collected simultaneously using computer vision.

Data Integration and Cleaning

Data from ventilator and paramonitor is synced with respect to time stamp.

Any missing value are cleaned out and data is corrected manually in case of collection error.

Data Analysis

Exploratory Data Analysis (EDA) is performed to understand and interpret data better.

Techniques used: Analysis of Variance (ANOVA), K-means clustering, and t-distributed Stochastic Neighbor Embedding (t-SNE)

Model Training and Testing

Classification to detect cause and track progression of condition.

Evaluation: Accuracy, Precision, recall and F1 score

Results

Detected cause used to calculate criticality which can be used to suggest actions for every alarm.

Figure 13 Cause identification methodology

4.2.1. Identifying Data

The data structure and requirements are the same as in section 4.1.

4.2.2. Data Simulation and Acquisition

Gathering and capturing real world data for such solutions is the ideal choice. But this approach has lot of challenges. Lack of digital infrastructure in hospitals, ethical obstacles and unavailability of proper datasets are some of them. This prompted the data simulation idea.

4.2.2.1. Data Simulation

A Simulator, the testchest, is used to simulate various patient conditions and vital signs. And the responses are recorded from the ventilator and para monitor screens.

Few experiments were carried out to simulate various faults in ventilation while healthy and diseased patients are being ventilated. The experiment details are as follows:

Objectives:

- To form a dataset that can be used to detect cause, i.e., mechanical failures

- To capture data for leakages and blockages while running different patient lung conditions

Theory: Testchest and its use, specification parameters

Apparatus:

i. Testchest
ii. Ventilator
iii. Para monitor
iv. Patient circuit
v. Endotracheal tube with ID 7
vi. A simple open/shut valve with ID 7
vii. Air compressor

Experiment setup:

Testchest is the lung simulator and simulates vital signs. It is connected to a ventilator via ET tube and patient circuit. The patient circuit has two arms, one for inhalation and the other for exhalation connected to respective ports in ventilator.

The testchest simulated patient vitals are transmitted to para monitor via pulse oximeter.

The system at the bottom left corner of the image uses a software called TestChest Organis to design and send simulation instructions to the testchest.

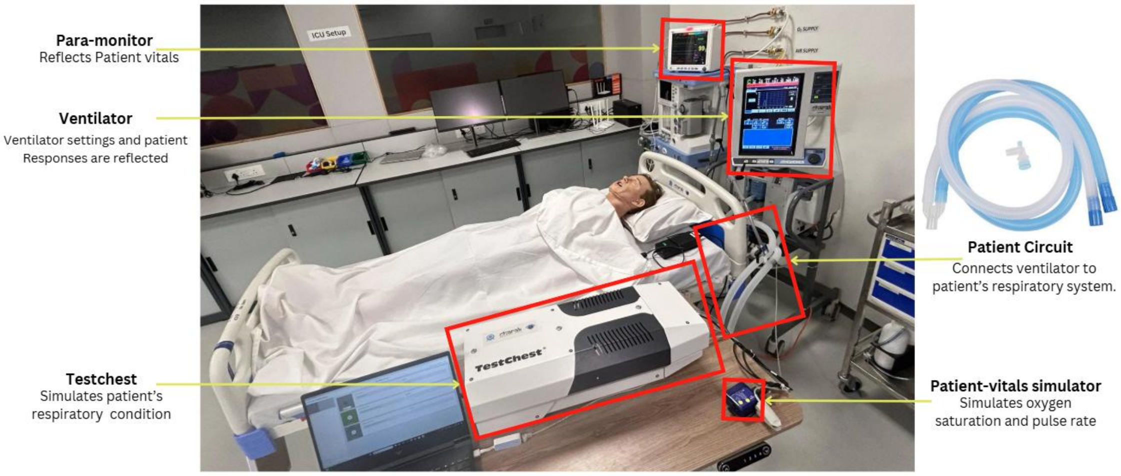

Para-monitor
Reflects Patient vitals
Ventilator
Ventilator settings and patient Responses are reflected
Testchest
Simulates patient's respiratory condition
Patient Circuit
Connects ventilator to patient's respiratory system.
Patient-vitals simulator
Simulates oxygen saturation and pulse rate
ICU Setup
TestChest

Figure 14 Experiment setup indication the experiment apparatus

Experiment control and response parameters:

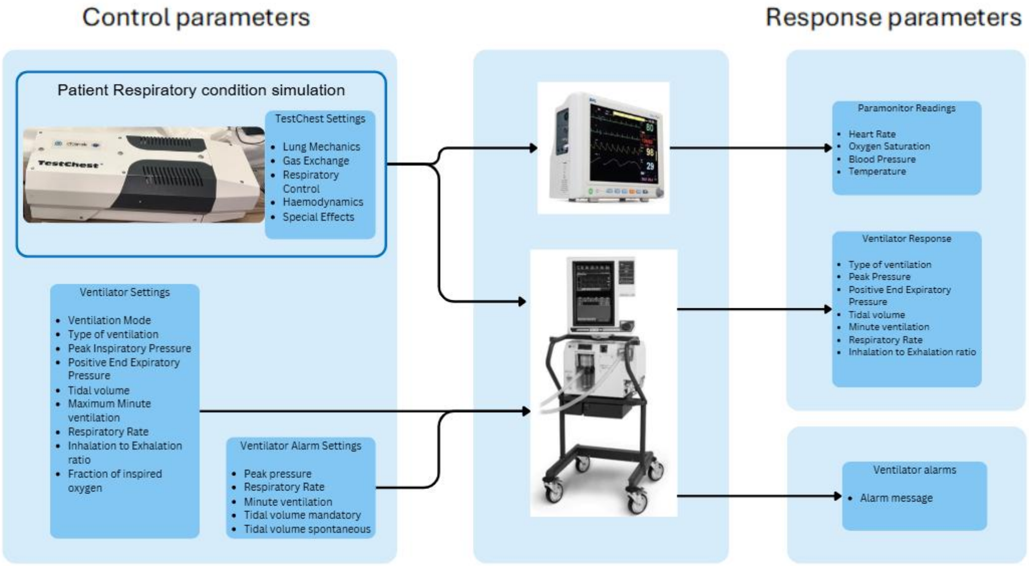


Figure 15 The setup information and response parameters of the experiment

The Control parameters have three main divisions. First, the patient's respiratory condition, is manipulated using 33 parameters classified in 5 broad categories- Lung Mechanics, Respiratory Control, Hemodynamic, Gas Exchange, and Special Effects. Secondly, the ventilator settings, to ventilate the simulated lung. And finally, ventilator alarm settings, set to trigger alarms based on ventilation requirements.

Response parameters are the simulation results that are reflected on the ventilator and para monitor screens.

The Control parameter settings for 2 experiments are as follows:

Table 3 Experiment 01 control parameters for healthy and Severe Asthma patients

Experiment 01 for Healthy and Severe Asthma patients			
Patient details			
Age :		**Adult >18**	**Adult >18**
Weight :		**65**	**65**
Patient Lung condition :		Healthy	Severe Asthma
Ventilator Settings			
Mode:		A/C	A/C
Peak Inspiratory Pressure:	cm H2O	VC	VC
TV :	ml	470	470
MV :	L/min	5.64	5.64
RR :	frequency	12	12
PEEP :	cm H2O	3	3
I:E		01:02	01:02
Vmin max	L/min	32	32
Vmin sens	L/min	3.3	3.3
PI	cmH2O	NA	NA
Ti	sec	NA	NA
Testchest settings			
Airway Resistance Raw		Rp5	Rp200
Collapse	%	0	30
Vdaw- dead airway		None	Large
Chest compliance, CW	mL/cm H2O	100	93
FiO2	%	30	80

Table 4 Experiment 02 control parameters for healthy and VAP patients

Experiment 02 for Healthy and Ventilator Associated Pneumonia patients					
Patient details					
Age :		**Adult >18**	**Adult >18**	**Adult >18**	**Adult >18**
Weight :		**65**	**65**	**65**	**65**

Patient Lung condition :		Healthy	Mild VAP	Moderate VAP	Severe VAP
		Ventilator Settings			
Mode:		A/C	A/C	A/C	A/C
Peak Inspiratory Pressure:	cm H2O	VC	VC	VC	VC
TV :	ml	450	450	450	450
MV :	L/min	5.85	5.85	5.85	5.85
RR :	frequency	13	13	13	13
PEEP :	cm H2O	3	3	3	3
I:E		01:02	01:02	01:02	01:02
Vmin max	L/min	32	32	32	32
Vmin sens	L/min	3	3	3	3
PI	cmH2O	NA	NA	NA	NA
Ti	sec	NA	NA	NA	NA
		Testchest settings			
Airway Resistance Raw		Rp5	Rp5	Rp20	Rp50
Collapse	%	0	10	30	60
Lung compliance, Cl	mL/cm H2O	117	90	70	35
Chest compliance, CW	mL/cm H2O	93.6	94	94	94
Total compliance, Crs	mL/cm H2O	52	42	31	21
FiO2	%	30	40	60	90

Table 5 Alarm thresholds set on ventilator during experimentation

Alarm Thresholds			
	Lower limit	Upper limit	
Ppeak	-	40	cm H2O
Ftot	-	30	frequency
Vminetot	3.25	6	L/min
Vtemand	330	610	ml
Vte spont	330	610	ml

All the experiments are done for adult patients that need mandatory ventilation.

These parameters are selected based on discussions with doctors and anesthesiologists.

Experiment scenarios:

Experiment 01:

Table 6 Faults induced and the method to simulate it in Experiment 01

Disease	Induced Fault	Simulates
Healthy	Healthy	No fault
	LTCleakage	Large leakage due to ET tube using testchest
	MTCleakage	Medium leakage due to ET tube using testchest
	70Blockage at IS due to pipe squeeze	70% patient circuit blockage on Inhalation arm due to kink
	95Blockage at IS due to pipe squeeze	95% patient circuit blockage on Inhalation arm due to kink
	70Blockage at ES due to pipe squeeze	70% patient circuit blockage on exhalation arm due to kink
	95Blockage at ES due to pipe squeeze	95% patient circuit blockage on exhalation arm due to kink
	STCleakage	Small leakage due to ET tube using testchest
	ISL1Blockage	30.67% patient circuit blockage using zip tie-diameter reduction at Inhalation side
	ISL2Blockage	45.33% patient circuit blockage using zip tie-diameter reduction at Inhalation side
	ESL3Blockage	24% patient circuit blockage using zip tie-diameter reduction at exhalation side
	ESL4Blockage	33.33% patient circuit blockage using zip tie-diameter reduction at exhalation side
	ESL5Blockage	56% patient circuit blockage using zip tie-diameter reduction at exhalation side
	EScCutleakage	2cm cut on patient circuit arm-Leakage in exhalation arm closer to ET tube
	IScCutleakage	2cm cut on patient circuit arm- Leakage in inhalation arm closer to ET tube
	ESACutleakage	2cm cut on patient circuit arm- Leakage in exhalation arm away to ET tube
	ISACutleakage	2cm cut on patient circuit arm- Leakage in inhalation arm away to ET tube
	Inside tube blockage	Blockage using a cotton ball inside the patient circuit
Severe Asthma	Normal	No fault
	LTCleakage	Large leakage due to ET tube using testchest
	MTCleakage	Medium leakage due to ET tube using testchest
	STCleakage	Small leakage due to ET tube using testchest
	ISAleakage	2cm cut on patient circuit arm- Leakage in inhalation arm away to ET tube
	ESAleakage	2cm cut on patient circuit arm- Leakage in exhalation arm away to ET tube

ESCleakage	2cm cut on patient circuit arm-Leakage in exhalation arm closer to ET tube
ISCleakage	2cm cut on patient circuit arm- Leakage in inhalation arm closer to ET tube
E180tubekink	Blockage due to 180 degree tube kink at exhalation arm
I180tubekink	Blockage due to 180 degree tube kink at inhalation arm

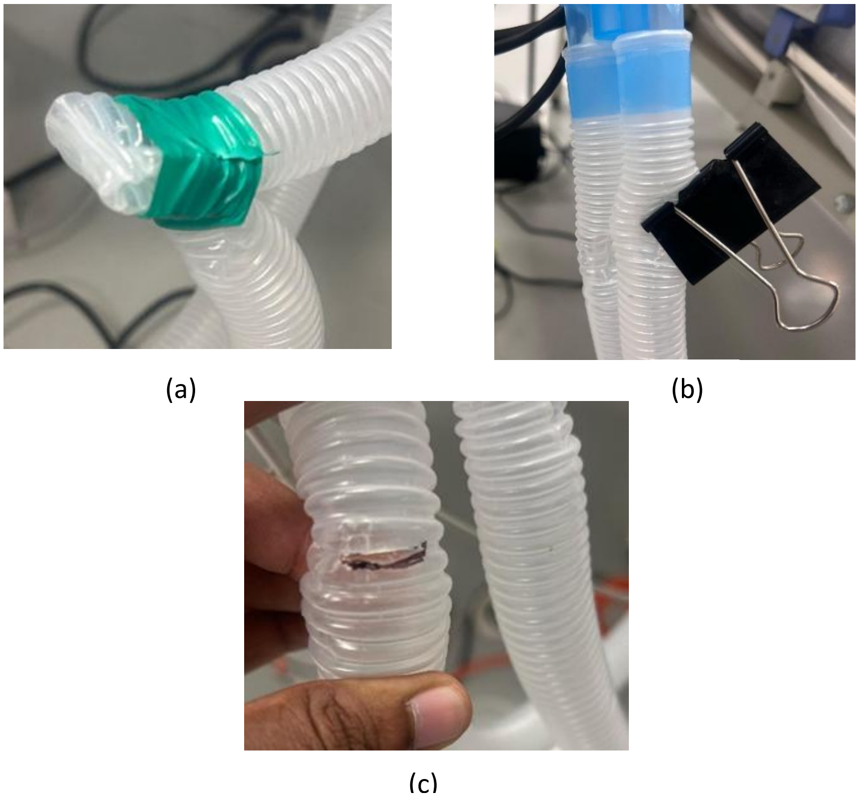


Figure 16 a)180 degrees Patient circuit kink, b) Simulating blockage using patient circuit squeeze c) Leakage simulation using a cut in patient circuit

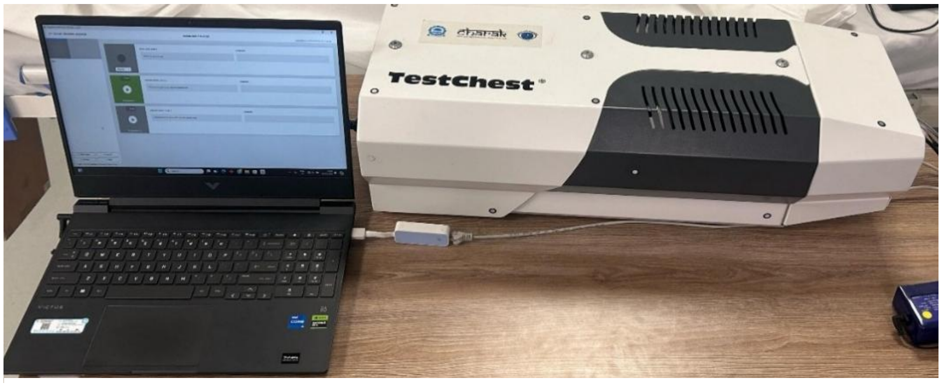


Figure 17 Testchest used for leakage simulation

Experiment 02:

Experiments for three faults or causes at different levels while simulating diseased lung conditions are performed in the following manner-

Table 7 Induced fault and method of simulation for Experiment 02

Disease	Induced Fault	Simulates
Healthy	Normal	No fault
	STCleakage	leakage due to ET tube
	MTCleakage	leakage due to ET tube
	LTCleakage	leakage due to ET tube
	L1ETTblockage	Obstruction inside ET tube
	L2ETTblockage	Obstruction inside ET tube
	L3ETTblockage	Obstruction inside ET tube
	20DETTblockage	ET Tube Kink/ bend
	25DETTblockage	ET Tube Kink/ bend
	30DETTblockage	ET Tube Kink/ bend
VAP-mild	Normal	No fault
	STCleakage	leakage due to ET tube
	MTCleakage	leakage due to ET tube
	LTCleakage	leakage due to ET tube
	L1ETTblockage	Obstruction inside ET tube
	L2ETTblockage	Obstruction inside ET tube
	L3ETTblockage	Obstruction inside ET tube
	20DETTblockage	ET Tube Kink/ bend
	25DETTblockage	ET Tube Kink/ bend
	30DETTblockage	ET Tube Kink/ bend
VAP-moderate	Normal	No fault
	STCleakage	leakage due to ET tube
	MTCleakage	leakage due to ET tube
	LTCleakage	leakage due to ET tube
	L1ETTblockage	Obstruction inside ET tube
	L2ETTblockage	Obstruction inside ET tube
	L3ETTblockage	Obstruction inside ET tube
	20DETTblockage	ET Tube Kink/ bend
	25DETTblockage	ET Tube Kink/ bend
	30DETTblockage	ET Tube Kink/ bend
VAP-Severe	Normal	No fault
	STCleakage	leakage due to ET tube

	MTCleakage	leakage due to ET tube
	LTCleakage	leakage due to ET tube
	L1ETTblockage	Obstruction inside ET tube
	L2ETTblockage	Obstruction inside ET tube
	L3ETTblockage	Obstruction inside ET tube
	20DETTblockage	ET Tube Kink/ bend
	25DETTblockage	ET Tube Kink/ bend
	30DETTblockage	ET Tube Kink/ bend

The following table explains the method used to simulate the faults.

Induced Fault	Method to achieve fault
Normal	No induced fault
STCleakage	Small leakage through test chest settings
MTCleakage	Medium leakage through test chest settings
LTCleakage	Large leakage through test chest settings
L1ETTblockage	47.79 % blockage of ET tube opening
L2ETTblockage	69.90 % blockage of ET tube opening
L3ETTblockage	91.16 % blockage of ET tube opening
20DETTblockage	20 degrees valve position
25DETTblockage	25 degrees valve position
30DETTblockage	30 degrees valve position

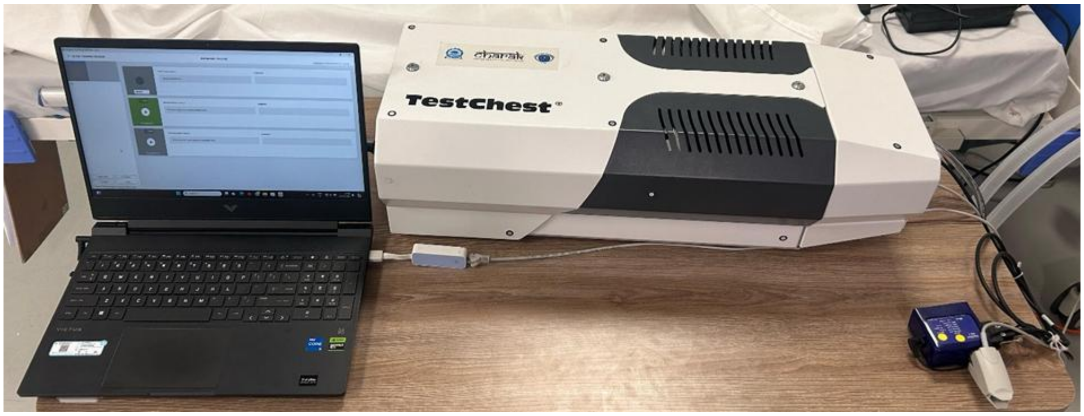


Figure 18 Testchest used for leakage simulation

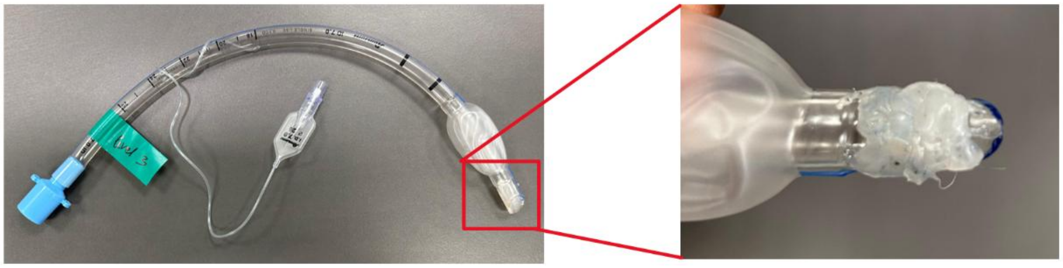


Figure 19 Simulation of in-tube blockage obstruction inside ET tube

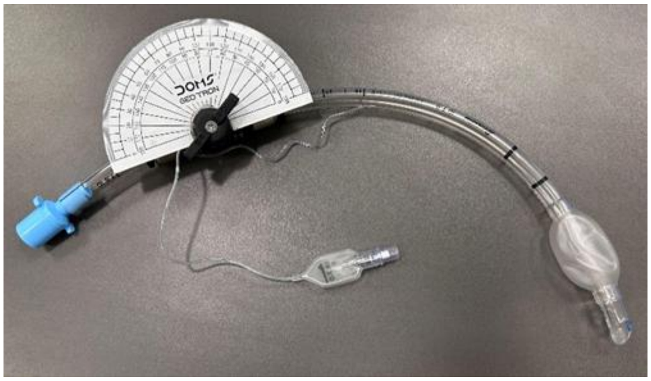


Figure 20 Simulation of blockage of ET Tube kink/bend using a valve

Experiment procedure:

i. Design lung condition parameters from literature and verify with doctors.

ii. Follow the standard procedures to start and set up the individual devices.

iii. Feed the lung parameters to testchest via TestChest Organis and start the simulation.

iv. Connect the para monitor to the testchest using a pulse oximeter.

v. Connect the patient circuit to the ventilator and at the other end with single opening, connect the ET tube.

vi. Set up the ventilator using the decided control parameters and start ventilation.

vii. Insert the open end of ET tube in the testchest opening and inflate the cuff to hold it in place.

viii. The setup is complete. Ventilator and para monitor screens will show the responses.

ix. Run different fault scenarios and capture data.

x. Once done with fault scenarios, change the patient condition using test chest and repeat the fault scenarios.

4.2.2.2. Data Acquisition

Data Acquisition: This stage encompasses the methods used to collect raw data from various sources. This could involve sensor reading, scraping web data, or gathering information through surveys and application logs.

The experiments generate the necessary data. This data needs to be collected in a digital format. The best way to capture this data would be to take data directly from the devices, but there are ethical restrictions to this method.

So, an optical character recognition solution was used to capture data from the device screens.

A camera focuses on the device screen and captures images alternatively for both the devices. The clicked image is then processed using an Optical character recognition (OCR) model. This model uses YOLOv7 for object detection and QWEN for character recognition. The extracted data from the images is stored in Excel files. These files are further processed to get data in the required format.

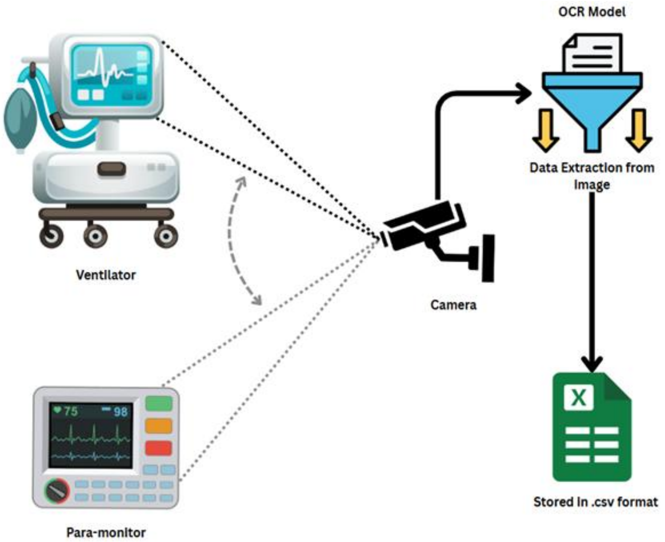


Figure 21 Data Acquisition method

2.3. Data Collection and integration

Data collected in excel format is stored in multiple files. Ventilator and para monitor data are stored in two different files. While the third file contains the labels for all the causes and the timestamps for which each experiment was run.

All data from these three files was compiled in one using the timestamps as the connecting link. This step is called data integration.

4.2.4. Data Preprocessing

Data preprocessing involves cleaning, transforming, and organizing raw data to make it suitable for analysis or model training. It's a crucial step in data science and machine learning, enhancing data quality, ensuring consistency, and preparing data for specific analytical techniques.

The raw data obtained needs preprocessing to get it ready to perform any analysis on it. In this step, all the missing values and duplicates are dealt with using multiple methods.

The integrated data from the experiments have multiple missing values. The reason being not every datapoint in the Ventilator and para monitor file had the linking time stamp in the label file, as the data was collected in a continuous manner, each scenario in the experiments had some stabilizing time.

Some data point had missing value, while others had wrong entries in wrong columns. These were handled by setting the threshold for the columns and the odd values were replaced with the column mean of that scenario.

outliers along the whole dataset.

4.2.5. Exploratory data analysis (EDA)

EDA is performed to extract trends and correlation among parameters from the data. It is performed in the following steps:

Exploration:

- **Univariate Analysis:** Analyse individual variables using descriptive statistics (mean, median, mode, range, variance, standard deviation) and visualizations (histograms, box plots).
- **Bivariate Analysis:** Examine relationships between two variables using scatter plots, heatmaps, or other visualizations.
- **Multivariate Analysis:** Investigate interactions between multiple variables, often using correlation analysis.
- **Multivariable analysis:** one outcome but multiple predictor (independent) variables (e.g., multiple regression, logistic regression with several covariates).

The aim is to identify cause using multiple variables. So, a multivariable ANOVA analysis was performed to find out the significance of each variable on outcome.

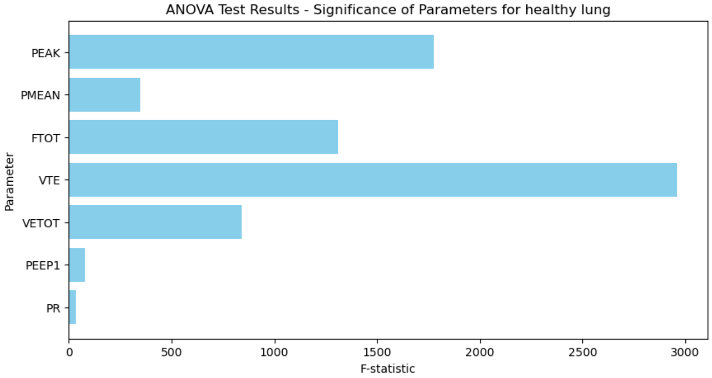


Figure 22 ANOVA results for healthy patient data in Experiment 01

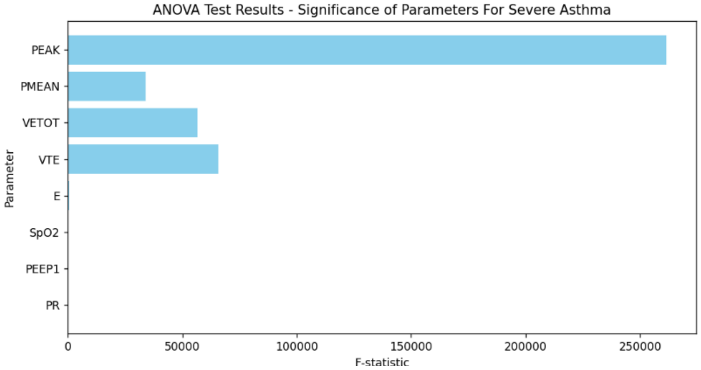


Figure 23ANOVA results for Severe Asthma patient data in Experiment 01

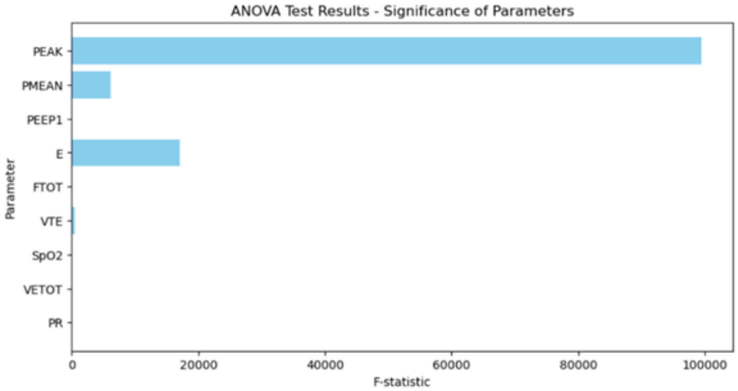


Figure 24 ANOVA results for Experiment 02

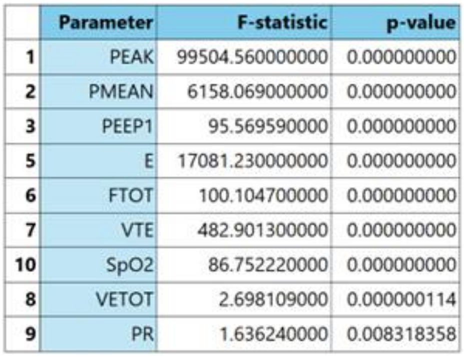
Feature Engineering:

Most prominent and relevant features based on the ANOVA results were selected for model training.

Hypothesis Testing:

Whether the collected parameters have any significant change with respect to the related causes. So, let the null hypothesis be, parameters have no significant impact on the outcome. And alternate hypothesis be, parameters have significant impact on the outcome

Table 8 F and P values from ANOVA Results for Experiment 02



	Parameter	F-statistic	p-value
1	PEAK	99504.560000000	0.000000000
2	PMEAN	6158.069000000	0.000000000
3	PEEP1	95.569590000	0.000000000
5	E	17081.230000000	0.000000000
6	FTOT	100.104700000	0.000000000
7	VTE	482.901300000	0.000000000
10	SpO2	86.752220000	0.000000000
8	VETOT	2.698109000	0.000000114
9	PR	1.636240000	0.008318358

Conclusion:

Ppeak, VTE, Pmean, PEEP1, E and VETOT are the parameters that have significant p values in ANOVA test, as can be seen from figures 24 and 25 for experiment 01 and figure 26 and table 8 for experiment 02. So, we reject null hypothesis and accept the alternate.

So, these parameters are selected to train the ML models for cause identification.

4.2.6. Models training and testing

The problem at hand is to identify the class to which the incoming parameters belong. So, this is a classification problem.

9 Classification models were trained and tested on the data generated. While training the models, 5-fold cross validation and hyperparameter tuning was carried out using GridSearchCV. These techniques ensured the reliability of the accuracies obtained from testing.

Results are discussed in the results and discussions section.

4.3. Mapping alarms, causes and responsive actions

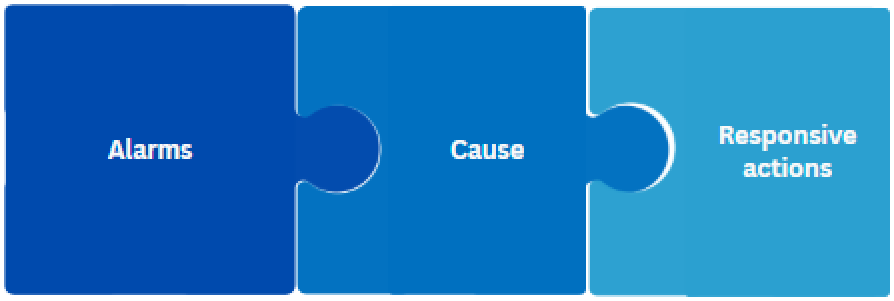


Figure 25 Relationship mapping between Alarm, cause and Responsive action

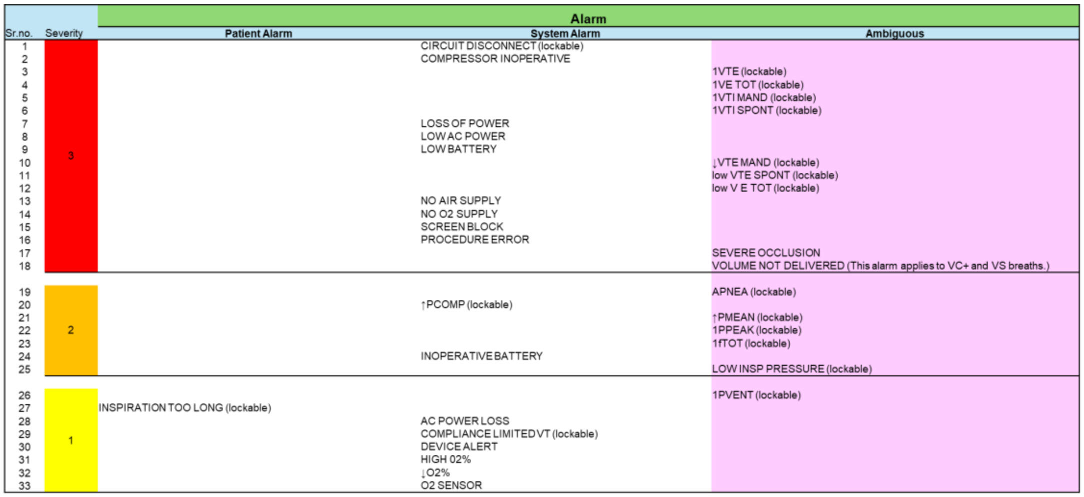
Establishing a clear correlation between alarms and causes helps narrow down the recommended action suggestions. Segregation of ambiguous alarms into patient and system alarms is possible as the cause can reflect that information.

Below are 3 causes that are used to segregate alarms as patient and system alarms.

Table 9 Simulated Cause and Alarm type

Alarm type	Cause
Patient	Obstruction inside ET tube
System	ET Tube Kink/ bend
	Leakage due to ET tube

Table 10 Risk factor associated with the alarm and classification of patient and system alarm for PB 840 ventilator



		Alarm		
Sr.no.	Severity	Patient Alarm	System Alarm	Ambiguous
1	3		CIRCUIT DISCONNECT (lockable)	
2			COMPRESSOR INOPERATIVE	
3				1VTE (lockable)
4				1VE TOT (lockable)
5				1VTI MAND (lockable)
6				1VTI SPONT (lockable)
7			LOSS OF POWER	
8			LOW AC POWER	
9			LOW BATTERY	
10				↓VTE MAND (lockable)
11				low VTE SPONT (lockable)
12				low V E TOT (lockable)
13			NO AIR SUPPLY	
14			NO O2 SUPPLY	
15			SCREEN BLOCK	
16			PROCEDURE ERROR	
17				SEVERE OCCLUSION
18				VOLUME NOT DELIVERED (This alarm applies to VC+ and VS breaths.)
19	2			APNEA (lockable)
20			↑PCOMP (lockable)	
21				↑PMEAN (lockable)
22				1PPEAK (lockable)
23				1fTOT (lockable)
24			INOPERATIVE BATTERY	
25				LOW INSP PRESSURE (lockable)
26	1			1PVENT (lockable)
27		INSPIRATION TOO LONG (lockable)		
28			AC POWER LOSS	
29			COMPLIANCE LIMITED VT (lockable)	
30			DEVICE ALERT	
31			HIGH 02%	
32			↓O2%	
33			O2 SENSOR	

This table classifies alarms into 3 categories and labels each of them with some risk or severity. The third category represents the alarms that cannot directly be classified as patient or system alarms, more information about their cause is required to do so.

The table below correlated all the ambiguous alarms for PB840 with the causes to determine the type of alarm.

Table 11 Mapping correlation between alarm and cause

Alarm type	Sr. No.	Alarm	Cause (detected using ML model)
Patient	2	APNEA (lockable)	Obstruction inside ET tube
Patient	9	↑PMEAN (lockable)	Obstruction inside ET tube
Patient	10	↑PPEAK (lockable)	Obstruction inside ET tube
Patient	11	↑PVENT (lockable)	
Patient	12	↑VTE (lockable)	Patient spontaneous breathing
Patient	13	↑VE TOT (lockable)	Patient spontaneous breathing
Patient	14	↑VTI MAND (lockable)	Patient spontaneous breathing

Patient	15	↑VTI SPONT (lockable)	Patient spontaneous breathing
Patient	16	↑fTOT (lockable)	Patient spontaneous breathing
Patient	24	↓VTE MAND (lockable)	Obstruction inside ET tube
Patient	25	low VTE SPONT (lockable)	Obstruction inside ET tube
Patient	26	low V E TOT (lockable)	Obstruction inside ET tube
Patient	32	SEVERE OCCLUSION	
Patient	33	VOLUME NOT DELIVERED (This alarm applies to VC+ and VS breaths.)	
System	8	↑PCOMP (lockable)	ET Tube Kink/ bend
System	9	↑PMEAN (lockable)	ET Tube Kink/ bend
System	10	↑PPEAK (lockable)	ET Tube Kink/ bend
System	11	↑PVENT (lockable)	
System	12	↑VTE (lockable)	Leakage due to ET tube
System	13	↑VE TOT (lockable)	Leakage due to ET tube
System	14	↑VTI MAND (lockable)	Leakage due to ET tube
System	15	↑VTI SPONT (lockable)	Leakage due to ET tube
System	16	1fTOT (lockable)	
System	24	↓VTE MAND (lockable)	ET Tube Kink/ bend
System	25	low VTE SPONT (lockable)	ET Tube Kink/ bend
System	26	low V E TOT (lockable)	ET Tube Kink/ bend
System	32	SEVERE OCCLUSION	
System	33	VOLUME NOT DELIVERED (This alarm applies to VC+ and VS breaths.)	

Table 12 Correlating Cause and Responsive actions

Cause detected	**Responsive action suggestion**			
	Nurse	**Anaesthesiologist**	**Resident doctors**	**Senior Doctors/ Specialist**
No fault				
Leakage due to ET tube	Check leakages	Check ET tube cuff pressure and position		
	Check ET tube cuff pressure and position	Prescribe sedation in case of Agitated patient		

		Check Vitals and make necessary adjustments to ventilator settings		
Obstruction inside ET tube	Check for secretions in the ET tube	Check Vitals and make necessary adjustments to ventilator settings	Check patient condition	Check patient condition
	Check ET tube position		Prescribe medication	Change treatment course
	Suctioning		Call needed specialist	Call needed specialist
ET Tube Kink/ bend	Check ET tube position			
	Adjust patient head position			
	Check for Patient organs pressing or biting on ET tube			

These two tables connect alarms, causes and the responsive action suggestions together.

4.4. Criticality score calculation

The Criticality score takes two major factors into consideration. The first factor being the state of patient's vital signs while the second factor is the type of alarm and its associated risk towards the patient.

Formula:

$$Criticality\ Score = F_{Vitals} + \alpha * F_{Alarmtype} + \beta * (F_{Alarmseverity} - 1)$$

$$F_{Vitals} - Factor\ of\ criticality\ indicating\ patient\ vitals\ state$$

$$F_{Alarmtype} - Factor\ of\ criticality\ indicating\ type\ of\ alarm$$

$$F_{Alarmseverity} - Factor\ of\ criticality\ indicating\ alarm\ criticality$$

$$\alpha, \beta - Multiplier$$

$$\beta = 30, constant$$

This equation makes sure that the criticality can be directly interpreted from the score itself. The threshold value for MEWS is ≥5. Meaning, if the score is higher than 4 the patient’s health is in critical state. With this reference, the threshold for Modified MEWS is also set at ≥5.

Table 13 Criticality score determining whether the alarm is critical or not based on the patient condition

Alarm type	**Non-critical**		**Critical**		**Severity with score**
	Lower limit	Upper Limit	Lower limit	Upper Limit	
System Alarm	0	4	5	24	↓ Increases
	30	34	35	54	
	60	64	65	84	
Patient Alarm	100	104	105	124	
	130	134	135	154	
	160	164	165	184	

Calculating F_{Alarm}

With the intention to distinguish the Patient and system critical alarms, the multiplier is introduced. This will enable easy mapping of data from the real world.

For Patient alarms, $\alpha = 100,$ and $F_{Alarmtype} = 1$

For System alarms, $\alpha = 1$ and $F_{Alarmtype} = 0$

Determining suitable ICU scoring system to calculate F_{Vital}

Few ICU scoring methods, [51],were studied and determined the best possible method for this application. As can be seen from the table below, MEWS is the most suitable scoring method. It is simple, considers the most relevant parameters and is proven.

Table 14 ICU Scoring methods

Score	Purpose	Type	Parameters considered
APACHE II	Severity & mortality prediction	Numeric	Physiological, chronic health, age, diagnosis
APACHE III/IV	Improved mortality prediction	Numeric	APACHE II parameters + more detailed chronic illness
SOFA	Organ dysfunction assessment	Numeric (daily use)	Respiration, Coagulation, Liver, CVS, CNS, Renal
qSOFA	Rapid screening for sepsis risk	Numeric (simple)	Respiratory rate, altered mentation, BP
SAPS II / SAPS III	Mortality prediction	Numeric	Physiological variables, demographics
MEWS	Early deterioration prediction	Numeric (simple)	BP, HR, RR, Temp, LOC
MODS	Multi-organ dysfunction	Numeric (organ-based)	Respiratory, renal, hepatic, cardiovascular, CNS
Glasgow Coma Scale (GCS)	Consciousness assessment	Numeric	Eye, verbal, motor response
RASS	Sedation & agitation	Numeric (behavioural)	Sedation-agitation levels
PIM & PRISM	Paediatric severity prediction	Numeric	Paediatric ICU-specific parameters

The Modified Early Warning Score (MEWS), [52],is a simple, physiological score that may allow improvement in the quality and safety of management provided to surgical ward patients. The primary purpose is to prevent delays in intervention or transfer of critically ill patients.

The threshold value for MEWS is ≥5. Meaning, if the score is higher than 4 the patient's health is in critical state.

Table 15 Modified Early Warning Score

Score	3	2	1	0	1	2	3
Respiratory rate (min^{-1})	-	≤ 8	-	9–14	15–20	21–29	> 29
Heart rate (min^{-1})	-	≤ 40	41–50	51–100	101–110	111–129	> 129
Systolic BP (mmHg)	≤ 70	71–80	81–100	101–199	-	≥ 200	-
Urine output (ml/kg/h)	Nil	< 0.5	-	-	-	-	-
Temperature (°C)	-	≤ 35	35.1–36	36.1–38	38.1–38.5	≥ 38.6	-
Neurological	-	-	-	Alert	Reacting to voice	Reacting to pain	Unresponsive

This system still needs to be refined and modified further to meet the application on hand. The modifications and additions needed to the scoring system are discussed ahead.

Calculating F_{Vital}

The second factor considers a total of 8 parameters. They are given on the table below. The scoring system used is based on an established method called Modified Early Warning Score (MEWS).

Table 16 Vital Criticality Score, based on MEWS

Parameter	**Score = 3**	**Score = 2**	**Score = 1**	**Score = 0 (Normal*)**	**Score = 1**	**Score = 2**	**Score = 3**
Heart Rate (bpm)	-	≤40	41–50	51–100	101-110	111–129	≥130
Systolic BP (mmHg)	≤70	71–80	81-100	101-199	-	≥200	-
Diastolic BP (mmHg)	<60	-	61-75	75-80	81-89	-	≥90
Respiratory Rate (/min)	-	≤8	-	9–14	15–20	21–29	≥30
Temperature (°C)	-	≤35.0	35.1–36.0	36.1–38.0	38.1–38.5	–	≥38.6
ABG: pH	≤7.30	–	–	7.31–7.45	–	–	≥7.46
ABG:$PaCO_2$(mmHg)	≤20	21–25	26–34	35–45	46–55	56–65	≥66
ABG: PaO_2 (mmHg)	≤40	41–55	56–70	71–100	101–200	–	–

Normal- the * indicates that the range of the normal is variable from patient to patient. To solve this problem, a personalized baseline approach is proposed*

With a few adjustments to the MEWS method, a new scoring method for the criticality score is formulated. The adjustments made are as follows:

i. Nonrelevant parameters- Urine output and Neurological are removed.

ii. Few additional parameters like diastolic BP, ABG- PaO2, PaCO2 and pH are added to make the score more relevant to the application[53].

Personalized vitals baseline generation

Vital parameter normal ranges differ from patient to patient. This leads to the issue that a generalized value range for vital parameters may not depict the actual patient condition. To overcome this issue, the need for personalized baseline for every patient is necessary. So, when a patient is admitted to an ICU, using the initial vitals data to calculate normal ranges which will serve as personalized base lie for that patient. This will help reduce all the initial patient condition-based alarms, which need to be ignored as suggested by doctors. The ranges will update as the data keeps updating.

Chapter 5: Results and Discussions

5.1. Results

Disease Identification

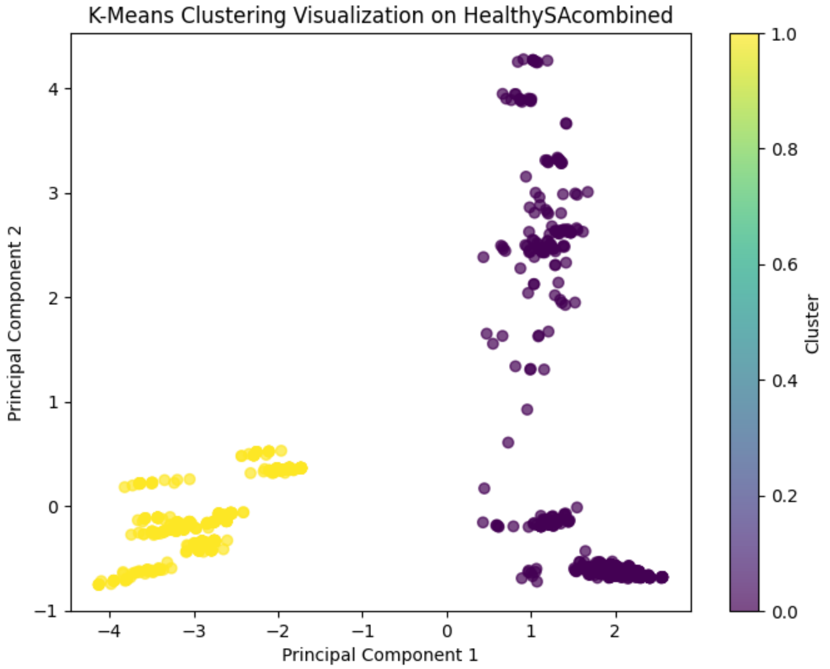
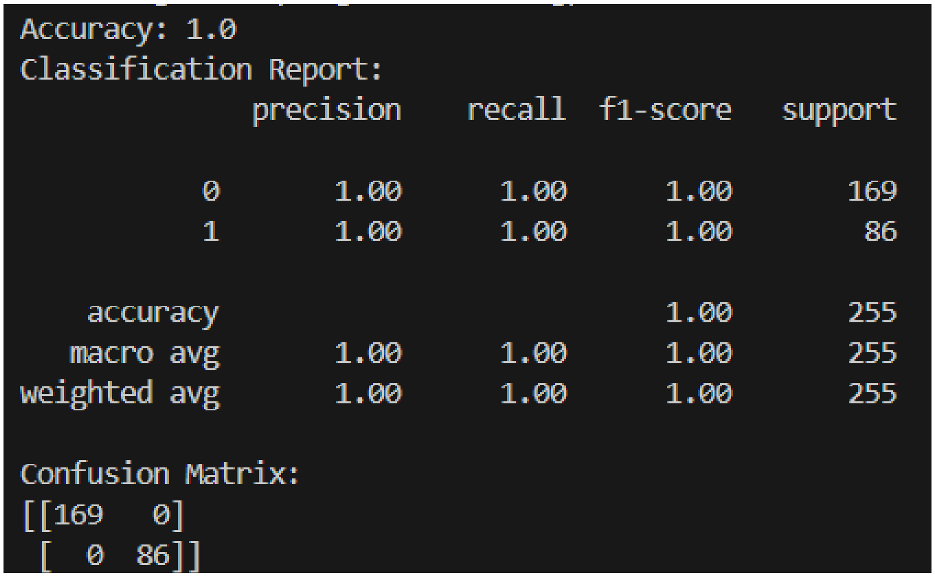


Figure 26 K-mean clustering to visualise the multi-dimensional dataset in 2D, Unsupervised learning to check the distinguishability in data groups



```
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       169
           1       1.00      1.00      1.00        86

    accuracy                           1.00       255
   macro avg       1.00      1.00      1.00       255
weighted avg       1.00      1.00      1.00       255

Confusion Matrix:
[[169   0]
 [  0  86]]
```

Figure 27 XGBoost Model Results for disease identification using Experiment 01 data

Both supervised and unsupervised learning models showed that the ventilator parameters can be used to identify diseases. Also, they serve as validation methods for the other learning methods.

Cause Identification Results

Experiment 01:

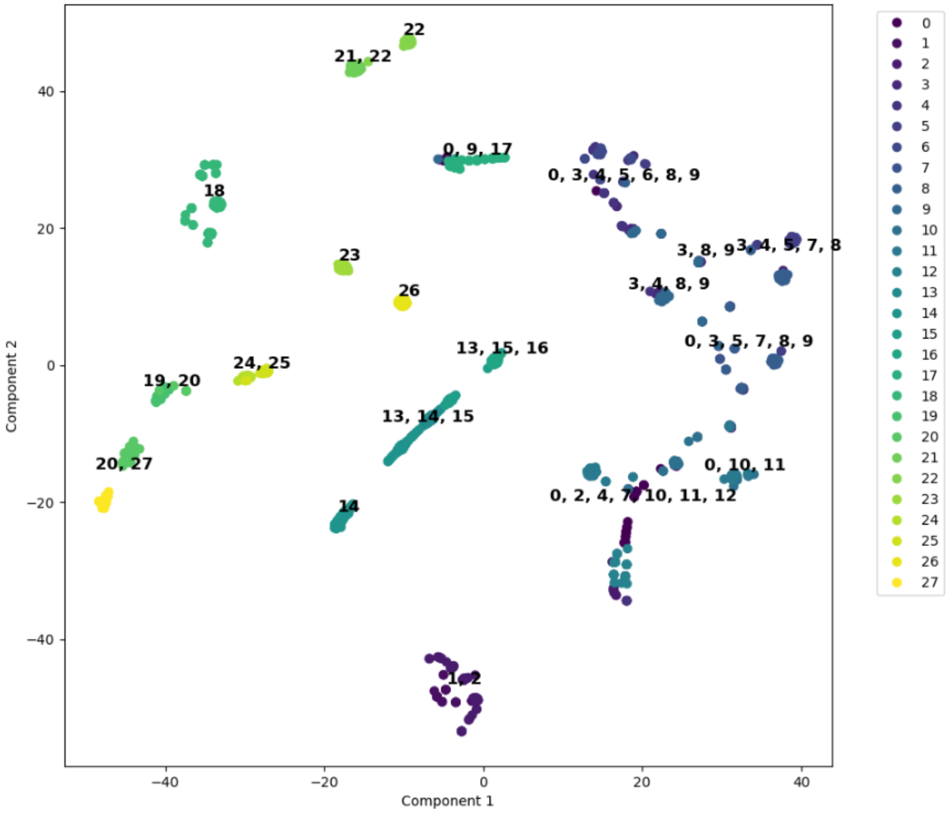


Figure 28 t-SNE plot visualising multiple clusters denoting to distinguishability in data group on Experiment 01 data

Table 17 Simulated disease and faults for Experiment 01

Label	Lung condition	Fault induced
0	**Healthy**	**None**
1	**Healthy**	**LTCleakage**
2	**Healthy**	**MTCleakage**

3	**Healthy**	**70Blockage at IS due to pipe squeeze**
4	**Healthy**	**90Blockage at IS due to pipe squeeze**
5	**Healthy**	**70Blockage at ES due to pipe squeeze**
6	**Healthy**	**90Blockage at ES due to pipe squeeze**
7	**Healthy**	**STCleakage**
8	**Healthy**	**ISL1Blockage**
9	**Healthy**	**ISL2Blockage**
10	**Healthy**	**ESL3Blockage**
11	**Healthy**	**ESL4Blockage**
12	**Healthy**	**ESL5Blockage**
13	**Healthy**	**EScCutleakage**
14	**Healthy**	**IScCutleakage**
15	**Healthy**	**ESACutleakage**
16	**Healthy**	**ISACutleakage**
17	**Healthy**	**Inside tube blockage**
18	**Severe Asthma**	**None**
19	**Severe Asthma**	**LTCleakage**
20	**Severe Asthma**	**MTCleakage**
21	**Severe Asthma**	**STCleakage**
22	**Severe Asthma**	**ISAleakage**
23	**Severe Asthma**	**ESAleakage**
24	**Severe Asthma**	**ESCleakage**
25	**Severe Asthma**	**ISCleakage**
26	**Severe Asthma**	**E180tubekink**
27	**Severe Asthma**	**I180tubekink**

The t-SNE plots are unsupervised models, these are used to check variation in the datapoints for different causes. The plot shows that similar data points making a small cluster showing these data points different from others. Each cluster is a different group of data points.

Some of the groups have multiple labels, this is either because the groups share similar characteristics or due to outliers. These problems can be solved with large quantity of data points.

Experiment 02:

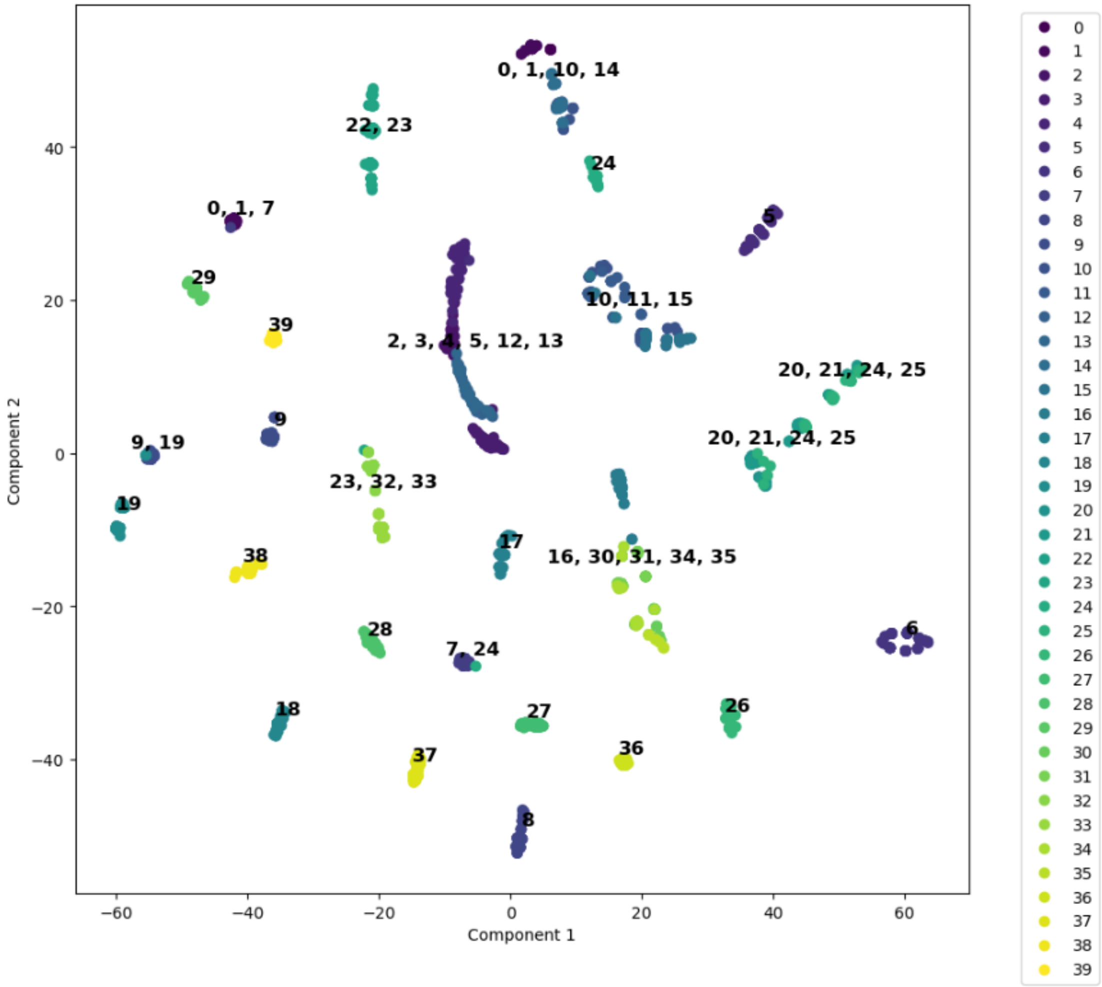


Figure 29 t-SNE visualisation for Experiment 02 data

Table 18 Faults induced for healthy and 3 levels of VAP patients

Disease	Induced_Fault	Simulates
Healthy	Normal	No fault
	STCleakage	leakage due to ET tube
	MTCleakage	leakage due to ET tube
	LTCleakage	leakage due to ET tube
	L1ETTblockage	Obstruction inside ET tube
	L2ETTblockage	Obstruction inside ET tube
	L3ETTblockage	Obstruction inside ET tube
	20DETTblockage	ET Tube Kink/ bend
	25DETTblockage	ET Tube Kink/ bend
	30DETTblockage	ET Tube Kink/ bend
VAP-mild	Normal	No fault
	STCleakage	leakage due to ET tube
	MTCleakage	leakage due to ET tube
	LTCleakage	leakage due to ET tube
	L1ETTblockage	Obstruction inside ET tube
	L2ETTblockage	Obstruction inside ET tube
	L3ETTblockage	Obstruction inside ET tube
	20DETTblockage	ET Tube Kink/ bend
	25DETTblockage	ET Tube Kink/ bend
	30DETTblockage	ET Tube Kink/ bend
VAP-moderate	Normal	No fault
	STCleakage	leakage due to ET tube
	MTCleakage	leakage due to ET tube
	LTCleakage	leakage due to ET tube
	L1ETTblockage	Obstruction inside ET tube
	L2ETTblockage	Obstruction inside ET tube
	L3ETTblockage	Obstruction inside ET tube
	20DETTblockage	ET Tube Kink/ bend
	25DETTblockage	ET Tube Kink/ bend
	30DETTblockage	ET Tube Kink/ bend
VAP-Severe	Normal	No fault
	STCleakage	leakage due to ET tube
	MTCleakage	leakage due to ET tube
	LTCleakage	leakage due to ET tube
	L1ETTblockage	Obstruction inside ET tube
	L2ETTblockage	Obstruction inside ET tube
	L3ETTblockage	Obstruction inside ET tube
	20DETTblockage	ET Tube Kink/ bend
	25DETTblockage	ET Tube Kink/ bend
	30DETTblockage	ET Tube Kink/ bend

This t-SNE plot shows that the groups of data points have distinct characteristics and can be easily distinguished.

Model Results:

Experiment 01:

Table 19 Classification Model results for Experiment 01

Model	Best Params	Accuracy	Precision	Recall	F1
Logistic Regression	{'C': 1, 'multi_class': 'multinomial', 'solver': 'lbfgs'}	0.7808	0.6746	0.7371	0.6931
Decision Tree	{'criterion': 'gini', 'max_depth': None, 'min_samples_split': 2}	0.9543	0.965	0.9582	0.9595
Random Forest	{'max_depth': None, 'n_estimators': 100}	0.9589	0.9639	0.9532	0.9544
k-Nearest Neighbors	{'n_neighbors': 3, 'weights': 'distance'}	0.9452	0.9549	0.9375	0.9388
MLP Neural Network	{'activation': 'tanh', 'alpha': 0.001, 'hidden_layer_sizes': (50,)}	0.863	0.8255	0.8588	0.8338
Support Vector Machine	{'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}	0.5525	0.2759	0.3888	0.3102
XGBoost	{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}	0.9498	0.9528	0.9341	0.9386
CatBoost	{'depth': 5, 'iterations': 100, 'learning_rate': 0.1}	0.9635	0.9731	0.963	0.9661
LightGBM	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 50}	0.9498	0.9522	0.9419	0.9437

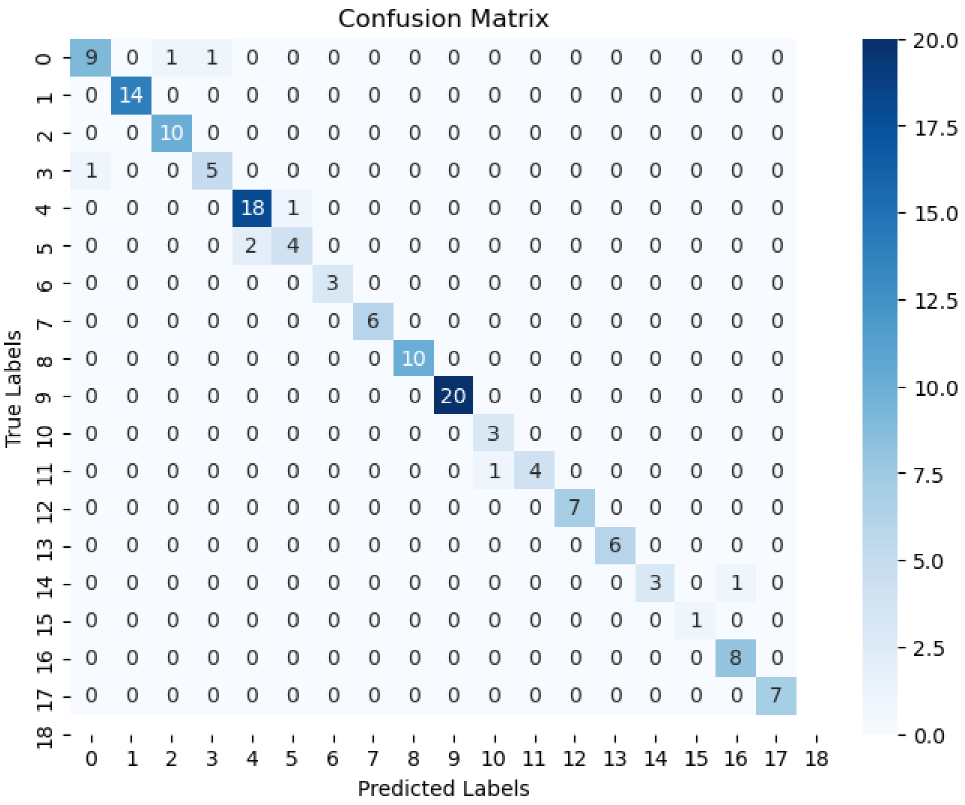


Figure 30 Confusion matrix of XGBoost for Experiment 01 data

Experiment 02:

Table 20 Classification Model results for Experiment 02

Model	Best Params	Accuracy	Precision	Recall	F1
Logistic Regression	{'C': 1, 'multi_class': 'multinomial', 'solver': 'lbfgs'}	0.5733	0.4079	0.5013	0.4219
Decision Tree	{'criterion': 'gini', 'max_depth': None, 'min_samples_split': 4}	0.9125	0.9117	0.9038	0.9038
Random Forest	{'max_depth': None, 'n_estimators': 50}	0.9125	0.9156	0.9059	0.9075
k-Nearest Neighbors	{'n_neighbors': 3, 'weights': 'uniform'}	0.8753	0.8753	0.8612	0.8636
MLP Neural Network	{'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (100,)}	0.6499	0.5039	0.5993	0.5216
Support Vector Machine	{'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}	0.1313	0.0175	0.0821	0.0277
XGBoost	{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}	0.9212	0.922	0.9157	0.916
CatBoost	{'depth': 5, 'iterations': 100, 'learning_rate': 0.1}	0.9125	0.9156	0.8949	0.8958
LightGBM	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 100}	0.9103	0.9088	0.8983	0.9004

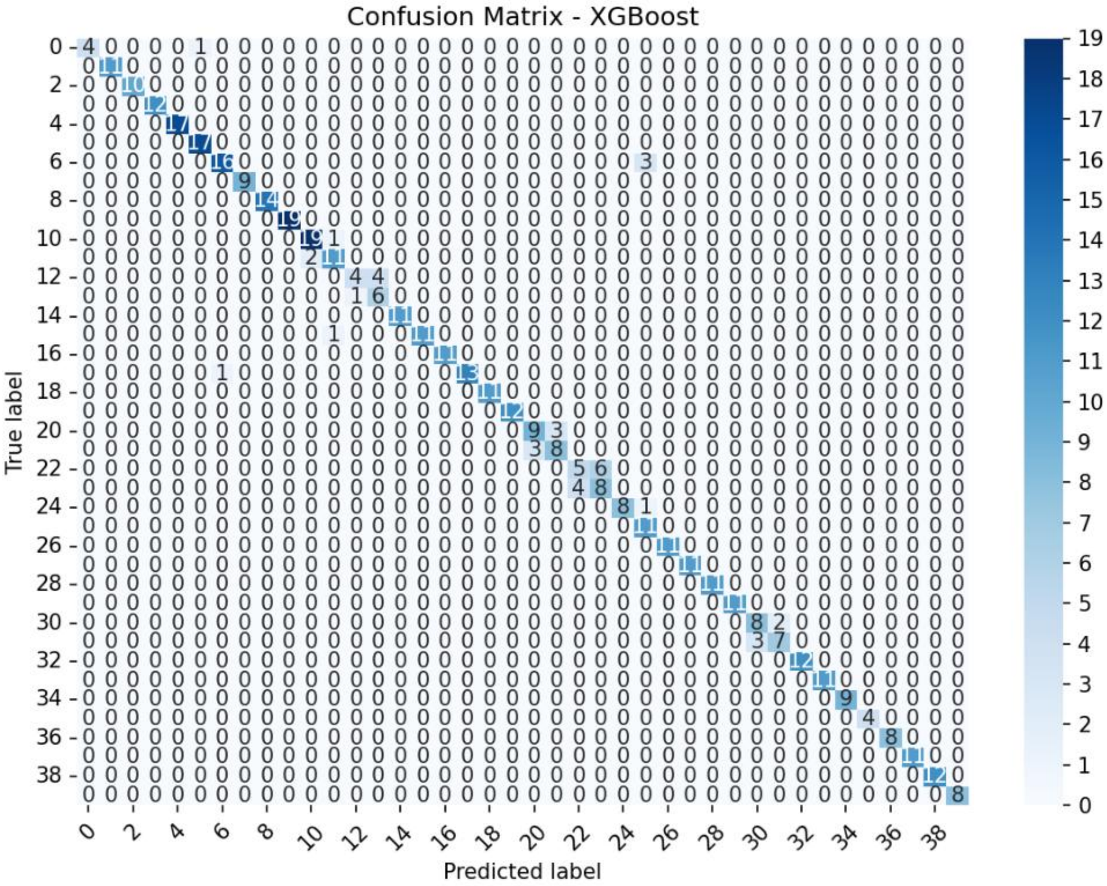


Figure 31 Confusion matrix of XGBoost for Experiment 02 data

In both cases, XGBoost performs best with the highest accuracy and F1 score.

This tells us that the faults can be identified from the dataset with 92 percent accuracy.

Example demonstrating the solution flow

Assuming a case of blockage in the ET tube due to patient secretions. Below example demonstrates the criticality calculation for the alarm(s) raised in this scenario.

Possible alarms raised:

i. High PPEAK
ii. Low VTE MAND
iii. Low V E TOT

Primary classification: Ambiguous alarm

Cause detected: Obstruction inside ET tube

Secondary classification: Patient Alarm

Criticality Calculation:

$$Criticality\ Score = F_{Vitals} + \alpha * F_{Alarmtype} + \beta * (F_{Alarmseverity} - 1)$$

$$F_{Alarm} = 1, for\ patient\ alarm\ from\ classification\ results$$

$$\alpha = 100$$

$$\beta = 30$$

$$F_{Alarmseverity} = 3 \quad [\textit{from Risk factor table- Table 10}]$$

Let the patient vitals be:

Table 21 Vitals table with associated score

	Values	Score
Heart Rate (bpm)	101	1
Systolic BP (mmHg)	135.5	0
Diastolic BP (mmHg)	62.5	1
Respiratory Rate (/min)	20	1
Temperature (°C)	39.61111111	3
ABG: pH	7.32	0
ABG:PaCO₂(mmHg)	39	0
ABG: PaO₂ (mmHg)	115	1
$\boldsymbol{F_{Vitals}}$	-	7

Now,

$$Criticality\ Score = F_{Vitals} + \alpha * F_{Alarmtype} + \beta * (F_{Alarmseverity} - 1)$$

$$Criticality\ Score = F_{Vitals} + 100 * F_{Alarmtype} + 30 * (F_{Alarmseverity} - 1)$$

$$Criticality\ Score = 7 + 100 * 1 + 30 * (3 - 1)$$

$$Criticality\ Score = 167$$

Hence, the alarm is critical, due to patient conditions and is triggered because of blockage inside ET tube, which is probably because of excessive secretions in the respiratory system of the patient.

The breakdown of the score is as follows:

i. F_{Vitals} being 7, which is greater than 4 (threshold for Modified-MEWS) which describes the poor vitals. (If the value were to be 4 or less, the score would be in between 160 and 164. Then the alarm would have been labelled as non-critical as the patient is stabilised.)

ii. $(\alpha * F_{Alarmtype})$= 100, as it is the patient alarm.

iii. $\left(\beta * \left(F_{Alarmseverity} - 1\right)\right)$=60, as the alarm risk factor is 3.

This shows that the criticality is controlled by the patient condition, as the thresholds set for alarm criticality are made keeping it as the anchor point.

5.2. Discussions

- Simulation settings and lung behavior remain constant for the entire duration of each scenario. It does not change dynamically once the scenario is running and needs to be changed manually for the next scenario. Although the behavior fluctuates a little to incorporate realistic data pattern.
- Each experiment is designed for a particular set of populations, adult population with complete respiratory support.
- As classification models are used, the classification happens amongst the existing classes, which means baseline for each disease is required so that an escalation pattern can be generalized once the patient condition is out of picture.
- The accuracy will improve with more data, as deep learning models perform best for these types of data where many variables are involved.

Chapter 6: Conclusion

Criticality of alarm is calculated based on patient condition and ventilator parameters can be used to detect the underlying cause.

Conclusions regarding cause Identification are:

- Ventilator parameters can be used to distinguish diseases.
- Faulty Patient circuit detection for leakage scenarios is successful. Different origins of leakages can be detected and classified.
- Blockage in Patient circuit due to tube twist does not pose an issue as its diameter is sufficiently large compared to smallest cross-section in the complete airway passage.
- In case of patient circuit tube blockage, alert is raised only at complete blockage.
- Halfway leakages and blockages are hidden in pressure and volume alarms.
- Subtle changes in diseases severity can be detected.
- Different types and levels of blockage can be identified just by analyzing the parameters.
- Blockage alarms for in-tube blockage are triggered at complete blockage but can be identified with the help of parameters.

The models developed for cause identification are 92% accurate. With this, it was observed that severe causes are more clearly identified as compared to minor faults. This works in favor of the solution.

A baseline for any disease is necessary to accurately access the faults, this is where the expert opinions of doctors can play a very important role. Similarly, personalized baseline of patient vitals is necessary to access the actual criticality of the alarm and fault rectification.

Chapter 7: Future scope

Building on existing alarm-management frameworks, future work should expand the catalogue of failure modes by systematically identifying additional causes of false or artifact-driven alarms—such as sensor dislodgement, tubing occlusions, or transient patient-ventilator desynchronies—and incorporate these into a unified detection strategy. Concurrently, time-series modelling techniques (e.g., ARIMA, LSTM, or Transformer architectures) can be developed to capture temporal patterns in ventilator parameters, enabling early detection of evolving faults rather than relying on fixed threshold breaches. By integrating these models into a prototype software platform, clinicians will gain real-time insights into device performance and patient status, minimizing nuisance alarms and improving clinical trust.

Once the software is built, targeted deployment in a controlled ICU or simulation lab will provide critical real-world feedback, allowing iterative refinement of algorithms and user interfaces. Data collected during this phase can seed a curated repository of annotated ventilator events, which will improve model robustness and generalizability. Finally, exploring additional data streams—such as waveform morphologies, patient-ventilator interaction metrics, or ventilator internal error logs—will further enhance the solution's ability to distinguish true alarms from benign artifacts, ultimately paving the way for a scalable, clinically validated alarm-management system.

References

[1] “HIGH-TECH STRATEGY 2020 FOR GERMANY.”

[2] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Business and Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, Aug. 2014, doi: 10.1007/s12599-014-0334-4.

[3] J. Lee, B. Bagheri, and H. A. Kao, “A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems,” *Manuf Lett*, vol. 3, pp. 18–23, Jan. 2015, doi: 10.1016/j.mfglet.2014.12.001.

[4] J. Wang, C. Xu, J. Zhang, and R. Zhong, “Big data analytics for intelligent manufacturing systems: A review,” *J Manuf Syst*, vol. 62, pp. 738–752, Jan. 2022, doi: 10.1016/j.jmsy.2021.03.005.

[5] E. Avdibasic, A. S. Toksanovna, and B. Durakovic, “Cybersecurity challenges in Industry 4.0: A state of the art review,” *Defense and Security Studies*, vol. 3, pp. 32–49, Aug. 2022, doi: 10.37868/dss.v3.id188.

[6] *Global Strategy on Digital Health 2020-2025*. World Health Organization, 2021.

[7] S. Abdulmalek *et al.*, “IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review,” Oct. 01, 2022, *MDPI*. doi: 10.3390/healthcare10101993.

[8] M. Elhaddad and S. Hamam, “AI-Driven Clinical Decision Support Systems: An Ongoing Pursuit of Potential,” *Cureus*, Apr. 2024, doi: 10.7759/cureus.57728.

[9] R. Krohn, “mHealth: a pathway to the intelligent hospital,” *Mhealth*, vol. 1, p. 16, 2015, doi: 10.3978/j.issn.2306-9740.2015.08.01.

[10] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[12] K. P. . Murphy, *Machine learning : a probabilistic perspective*. MIT Press, 2012.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge: MIT Press, 2016.

[14] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction Second edition, in progress.”

[15] F. Pedregosa FABIANPEDREGOSA *et al.*, “Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA,

VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," 2011. [Online]. Available: http://scikit-learn.sourceforge.net.

[16] T. Hastie, R. Tibshirani, and J. Friedman, "Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction."

[17] J. R. Quinlan, "Induction of Decision Trees," 1986.

[18] L. Breiman, "Random Forests," 2001.

[19] T. M. Cover and P. E. Hart, "Approximate formulas for the information transmitted bv a discrete communication channel," 1952.

[20] C. Cortes, V. Vapnik, and L. Saitta, "Support-Vector Networks Editor," Kluwer Academic Publishers, 1995.

[21] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[22] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features." [Online]. Available: https://github.com/catboost/catboost

[23] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." [Online]. Available: https://github.com/Microsoft/LightGBM.

[24] "PURITAN BENNETT 840 SERVICE MANUAL."

[25] "E X TEND-E X TEND XT User Manual." [Online]. Available: www.airliquide.comwww.airliquidemedicalsystems.com

[26] "https://www.drawellanalytical.com/patient-monitor-everything-you-should-know/#:~:text=the%20patient%20monitor.-,Patient%20Monitor%20Definition,%2C%20pulse%2C%20and%20body%20temperature."

[27] O. Gmbh, "TestChest V3 Instruction Handbook Keep document for future use."

[28] S. T. Holgate, "Pathogenesis of asthma.," Jun. 2008. doi: 10.1111/j.1365-2222.2008.02971.x.

[29] J. W. Mims, "Asthma: Definitions and pathophysiology," *Int Forum Allergy Rhinol*, vol. 5, pp. S2–S6, Sep. 2015, doi: 10.1002/alr.21609.

[30] J. Chastre and J.-Y. Fagon, “State of the Art Ventilator-associated Pneumonia,” *Am J Respir Crit Care Med*, vol. 165, pp. 867–903, 2002, doi: 10.1164/rccm.2105078.

[31] S. M. Koenig and J. D. Truwit, “Ventilator-associated pneumonia: Diagnosis, treatment, and prevention,” Oct. 2006. doi: 10.1128/CMR.00051-05.

[32] L. Papazian, M. Klompas, and C. E. Luyt, “Ventilator-associated pneumonia in adults: a narrative review,” May 01, 2020, *Springer*. doi: 10.1007/s00134-020-05980-0.

[33] V. Goel, S. A. Hogade, and S. G. Karadesai, “Ventilator associated pneumonia in a medical intensive care unit: Microbial aetiology, susceptibility patterns of isolated microorganisms and outcome,” *Indian J Anaesth*, vol. 56, no. 6, pp. 558–562, 2012, doi: 10.4103/0019-5049.104575.

[34] W. L. Jackson and A. F. Shorr, “Update in ventilator-associated pneumonia,” Lippincott Williams & Wilkins, 2006.

[35] O. Tronstad, D. Flaws, S. Patterson, R. Holdsworth, and J. F. Fraser, “Creating the ICU of the future: patient-centred design to optimise recovery,” *Crit Care*, vol. 27, no. 1, Dec. 2023, doi: 10.1186/s13054-023-04685-2.

[36] R. L. Shaw, R. Morrison, S. Webb, O. Balogun, H. P. Duncan, and I. Butcher, “Challenges to well-being in critical care,” *Nurs Crit Care*, vol. 29, no. 4, pp. 745–755, Jul. 2024, doi: 10.1111/nicc.13030.

[37] C. L. Cifra, J. W. Custer, and J. C. Fackler, “A Research Agenda for Diagnostic Excellence in Critical Care Medicine,” Jan. 01, 2022, *W.B. Saunders*. doi: 10.1016/j.ccc.2021.07.003.

[38] N. D. J. H. P. D. N. van der P. and M. de K. “A research agenda for diagnostic excellence in critical care,” Crit. C. vol. 26, Art. 10, 2022. A. Swaan, “Ventilator alarms: Challenges and Opportunities for Improvement,” 2017. [Online]. Available: http://aami-bit.org

[39] K. Ghorbanzadeh, A. Ebadi, M. Hosseini, S. S. B. Madah, and H. Khankeh, “Challenges of the patient transition process from the intensive care unit: A qualitative study,” *Acute and Critical Care*, vol. 36, no. 2, pp. 133–142, May 2021, doi: 10.4266/ACC.2020.00626.

[40] A. Konkani, B. Oakley, and B. Penprase, “Reducing Hospital ICU Noise: A Behavior-Based Approach.”

[41] S. Deb and D. Claudio, “Alarm fatigue and its influence on staff performance,” *IIE Trans Healthc Syst Eng*, vol. 5, no. 3, pp. 183–196, Jul. 2015, doi: 10.1080/19488300.2015.1062065.

[42] B. K. Walsh and J. B. Waugh, “Alarm strategies and surveillance for mechanical ventilation,” *Respir Care*, vol. 65, no. 6, pp. 820–831, Jun. 2020, doi: 10.4187/respcare.07546.

[43] “Managing Mechanical Ventilator Alarms with Middleware”.

[44] “Impact Of Clinical Alarms On Patient Safety,” 2006.

[45] S. Deb and D. Claudio, “Alarm fatigue and its influence on staff performance,” *IIE Trans Healthc Syst Eng*, vol. 5, no. 3, pp. 183–196, Jul. 2015, doi: 10.1080/19488300.2015.1062065.

[46] J. B. Scott, “The conundrum of mechanical ventilation alarms,” Apr. 01, 2021, *American Association for Respiratory Care*. doi: 10.4187/respcare.09028.

[47] B. Coldewey, A. Diruf, R. Röhrig, and M. Lipprandt, “Causes of use errors in ventilation devices - Systematic review,” *Appl Ergon*, vol. 98, Jan. 2022, doi: 10.1016/j.apergo.2021.103544.

[48] N. Asadi, F. Salmani, N. Asgari, and M. Salmani, “Alarm fatigue and moral distress in ICU nurses in COVID-19 pandemic,” *BMC Nurs*, vol. 21, no. 1, Dec. 2022, doi: 10.1186/s12912-022-00909-y.

[49] K. Kim, Y. Kim, Y. S. Kim, K. B. Kim, and S. H. Lee, “New Web-Based Ventilator Monitoring System Consisting of Central and Remote Mobile Applications in Intensive Care Units,” *Applied Sciences (Switzerland)*, vol. 14, no. 15, Aug. 2024, doi: 10.3390/app14156842.

[50] “ISO/IEC, Information Technology – Vocabulary – Part 1: Fundamental Terms, ISO/IEC 2382-1, 2015.,” 2015.

[51] T. P. Pellathy, M. R. Pinsky, and M. Hravnak, “Intensive care unit scoring systems,” *Crit Care Nurse*, vol. 41, no. 4, pp. 54–65, Aug. 2021, doi: 10.4037/ccn2021613.

[52] J. Gardner-Thorpe, N. Love, J. Wrightson, S. Walsh, and N. Keeling, “The Value of Modified Early Warning Score (MEWS) in Surgical In-Patients: A Prospective Observational Study,” *The Annals of The Royal College of Surgeons of England*, vol. 88, no. 6, pp. 571–575, Oct. 2006, doi: 10.1308/003588406X130615.

[53] P. S. Z. M. et al Castro D, “https://www.ncbi.nlm.nih.gov/books/NBK536919/,” StatPearls.