**BANL-6900 Business Analytics Capstone Final Project Report**

**Enhancing Alarm Intelligence through Machine Learning at OSUM Oil Sands**

**TEAM - 5**

Karthik Tulugu

Vamshi Krishna Kunchala

Sai Surya Kandela

Shashank Bharadwaj Asaram

Advisor: Dr. Mehmet Balcilar, Associate Professor

**University of New Haven**

A blue logo with text and a book

AI-generated content may be incorrect.

May 4, 2025

**Abstract**

This study examines chattering and alarm overload in OSUM Oil Sands operations, intending to improve alarm management using machine learning and data analytics. There are 20,000 records of alarm metadata in the dataset, such as chattering labels, activity durations, and tag types. Data cleansing, exploratory analysis, predictive modelling, and interactive visualization in R Shiny were all produced as part of a comprehensive workflow.

Key findings include identifying problematic alarm categories (“Others”) and time windows with frequent alarms. XGBoost emerged as the best-performing model for both classification (CHB detection, AUC = 0.91) and regression (ATD prediction). These insights support the deployment of intelligent alarm filtering to reduce operator overload and improve real-time decision-making.

These findings offer operational value by improving real-time alarm prioritization, reducing missed critical alerts, and supporting future deployment in real-time alarm streaming systems. An interactive Shiny dashboard demonstrating key visualizations and model performance is available at: <https://karthikkt001.shinyapps.io/alarmcasestudy>

**Introduction**

**Business Problem Identification**

Industrial alarm systems often suffer from information overload due to frequent, redundant, or ambiguous alerts, leading to alarm fatigue. This project addresses alarm management inefficiencies at OSUM Oil Sands by identifying, classifying, and predicting high-risk alarms, particularly those exhibiting chattering behavior (CHB). Our objective is to reduce operational disruptions and enhance situational awareness for operators through intelligent filtering. This project also aims to integrate advanced machine learning tools into operator workflows by using interpretable and scalable models supported by a real-time dashboard.

**Literature Review**

1. **Yang et al. (2015)** emphasized the need for alarm rationalization in industrial control systems. Their work highlighted how excessive and poorly managed alarms can lead to cognitive overload for operators, often resulting in missed critical events. This underscores the need for streamlined alarm filtering systems to improve decision-making.
2. **Qian et al. (2020)** demonstrated the utility of machine learning for detecting alarm floods and chattering behaviors. They applied supervised learning models to identify patterns associated with repeated alarm triggers, suggesting that automated detection can significantly reduce operator fatigue.
3. **Lee and Lee (2021)** explored the benefits of predictive modelling in prioritizing alarms. Their study found that using classification models to predict alarm urgency can help operators focus on high-impact events, thus reducing incident response time.
4. **Ma et al. (2018)** tested gradient boosting methods, including XGBoost, in complex industrial systems. They reported that XGBoost consistently delivered high accuracy and robustness across varied datasets, making it a strong candidate for alarm prediction tasks.
5. **Anderson et al. (2022)** talked about how R Shiny and other interactive visual analytics systems are used in operations. They underlined how well-designed dashboards may improve the usefulness and clarity of sophisticated analytics by bridging the gap between data scientists and field operators.

**Data Collection and Preparation**

**Dataset Details**

* Source: OSUM Oil Sands alarm logs (IM009B-XLS-ENG.xlsx)
* Three worksheets (Alarm Data – 20,000 rows x 20 columns, Test Data, Data for visualization)
* Key fields: Alarm Tag Type, ATD (Active Time Duration), CHB (Chattering), M, Week, H (Hour groups)
* Alarm Severity (Low, Medium, High – inferred if not explicit)
* M (Minute marker), H (Hour group), and Week used for temporal patterns

**Cleaning and Transformation**

* Missing values were removed using the `drop\_na()` function in R. All numeric fields were standardized, and categorical variables were converted into dummy variables for model compatibility.
* Converted CHB and ATD to numeric where necessary.
* Categorical encoding and model matrix preparation were applied before modelling.

**Methodology**

**Approach**

* Analytics Type: Predictive Analytics
* Task 1: Binary classification to predict chattering alarms (CHB = 1)
* Task 2: Regression to estimate Active Time Duration (ATD)

**Algorithms Used**

1. Logistic Regression – interpretable baseline for classification.
2. Decision Tree – handles nonlinear relationships.
3. XGBoost – advanced gradient boosting for performance.
4. Linear Regression – baseline model for ATD prediction.
5. All models were evaluated using 10-fold cross-validation on a 70:30 train-test split. Class imbalance was addressed using stratified sampling and performance metrics such as AUC and F1-score.

**Findings and Discussion**

**Exploratory Data Analysis (EDA) Highlights**

* ‘Level’ alarms are most frequent. Followed by ‘Flow’ and ‘Pressure’, ‘Others’ had ambiguity.

A diagram of a number of alarm tags

AI-generated content may be incorrect.

Figure 1: Distribution of Alarm Tag Types

This pie chart illustrates the relative frequency of different alarm tag types recorded in the dataset. The most common category is "Level" alarms, accounting for 35.9% of total alarms, followed by "Others" (21.2%), "Pressure" (16.1%), "Flow" (14.8%), and "Temperature" (11.9%). The visualization highlights the need to investigate and clarify ambiguous tags such as "Others."

A graph showing a number of green rectangular objects

AI-generated content may be incorrect.

Figure 5: Alarm Tag Type Frequency

This bar chart shows the count of alarms by tag type in the OSUM Oil Sands dataset. "Level" alarms are the most frequent, with over 6,000 occurrences, followed by "Others", "Pressure", "Flow", and "Temperature" in descending order. The high volume of "Others" underscores the need for improved tagging consistency and alarm categorization.

* 13–18-hour window showed alarm peaks – Suggesting operational bottlenecks.

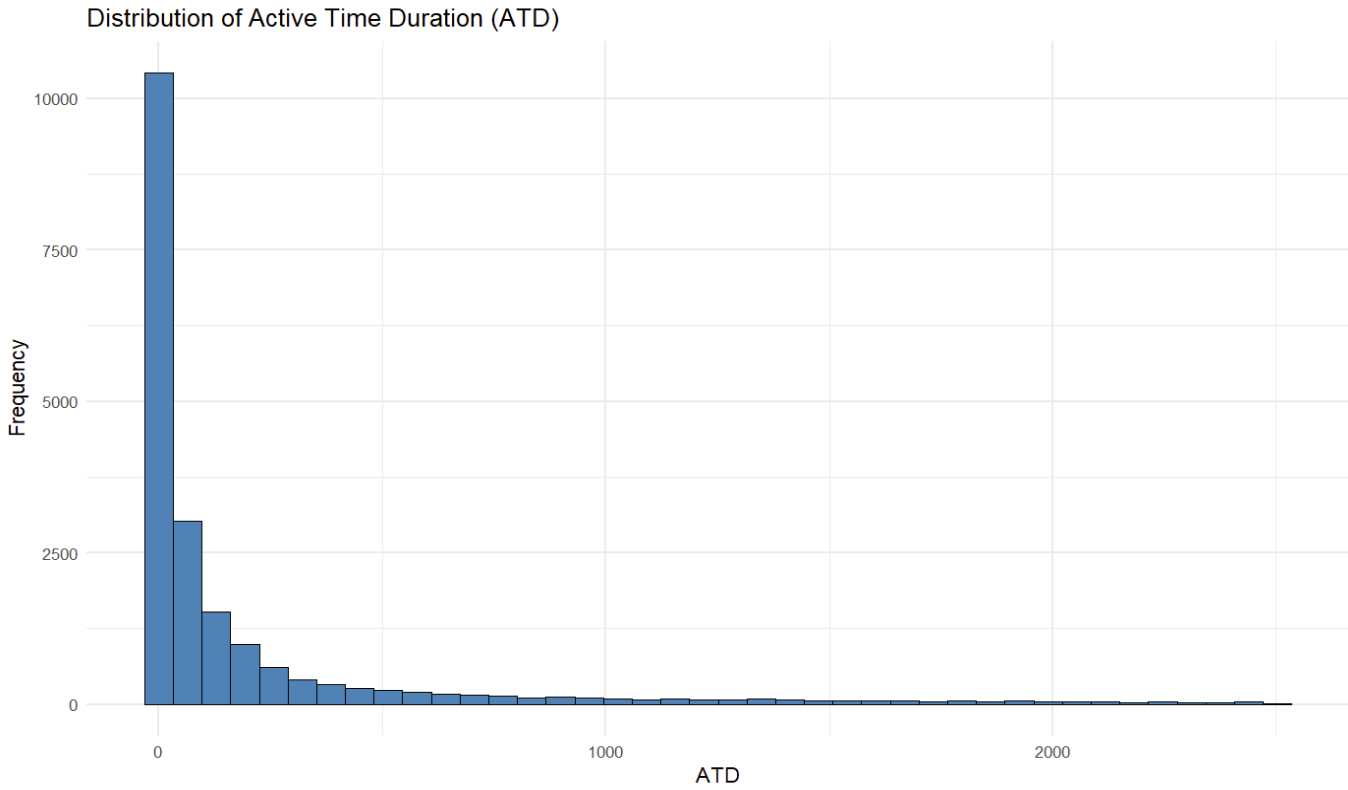


Figure 6: Distribution of Active Time Duration (ATD)

A graph of a bar chart

AI-generated content may be incorrect.

Figure 7: Total Alarm Counts by Hour Range

* CHB Behavior: 25% of alarms were chattering (**CHB=1**): This trend highlights the importance of label quality in alarm datasets and suggests a need for standardizing tag categories for effective classification.

A graph of a number of individuals

AI-generated content may be incorrect.

Figure 8: Distribution of Chattering Behavior (CHB)

* ‘Others’ Category: 21% of alarms in this category were chattering, indicating label quality issues.

A red and blue pie chart

AI-generated content may be incorrect.

Figure 9: CHB Distribution in ‘Others’ Alarm Type

**Classification Results (CHB):**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **AUC** |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression | 0.798 | 0.602 | 0.341 | 0.435 | 0.827 |
| Decision Tree | 0.838 | 0.678 | 0.553 | 0.609 | 0.847 |
| **XGBoost** | **0.865** | **0.722** | **0.663** | **0.691** | **0.913** |

A graph of different colored bars

AI-generated content may be incorrect.

Figure 10: CHB Classification – Model Performance Comparison

A graph of a model

AI-generated content may be incorrect.

Figure 11: ROC Curve Comparison – CHB Classification

**Regression Results (ATD)**

* Linear Regression: **RMSE = 407.5, MAE = 240.0**

A graph showing a plot of a graph

AI-generated content may be incorrect.

Figure 12: Actual vs Predicted ATD – Linear Regression

* XGBoost: **RMSE =** **404.0, MAE = 231.8**
* XGBoost demonstrated better handling of outliers and nonlinear relationships in alarm duration prediction compared to linear regression.A graph with dots and lines

  AI-generated content may be incorrect.

Figure 13: Actual vs Predicted ATD – XGBoost Regressor

* Residual plots confirmed XGBoost’s better fit and robustness across ATD values.

A graph of a graph showing a plot

AI-generated content may be incorrect.

Figure 14: Residual Plot – XGBoost Regressor

**Dashboard Visualizations**

To enhance model interpretability and provide real-time operational insights, we developed an interactive R Shiny dashboard. The following figures illustrate key components of the dashboard, including alarm metrics, classification model performance, and prediction accuracy. The final outputs, including visual insights and model comparison metrics, are deployed in an interactive dashboard accessible at <https://karthikkt001.shinyapps.io/alarmcasestu>dy

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 15:** Dashboard Overview

This figure presents the homepage of the Shiny dashboard, showing alarm metrics like total alarms, average flow, ATD distribution, and tag-wise alarm counts.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 16:** CHB Classification Models

Performance comparison of Logistic, Decision Tree, and XGBoost models for chattering alarm classification using key metrics.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 17:** ATD Predictions & Residuals

Visualization of actual vs. predicted ATD values and residuals using XGBoost and Linear Regression models.

**Application to the New Data**

The final XGBoost models can be applied to new streaming or batch alarm data collected daily from OSUM’s control systems to simulate deployment conditions. These models are optimized for batch and streaming data scenarios and can be easily integrated into OSUM’s daily alarm review pipelines or dashboard alert prioritization logic. For testing purposes, bootstrapped samples of the original dataset were used to verify the stability and generalizability of model predictions.

**Conclusion**

To address the increasing problem of alarm overload, this study used machine learning models to categorize and forecast alert behaviors. XGBoost was the top-performing model in both the regression and classification tests. Peak-hour alarm spikes and ambiguous “Others” category tags emerged as critical areas for improving operational efficiency and alarm categorization standards.

**Reflection**

* Smooth: Integration of modelling pipeline, strong collaboration across roles.
* Tools used included R, Shiny, XGBoost, caret, and tidyverse libraries, enabling an end-to-end pipeline from data wrangling to dashboard deployment.
* Challenges: Handling class imbalance; interpreting “Others” category alarms.
* Future Work: Expand to real-time streaming alarms, deeper classification into alarm root causes.

**References**

1. Yang, H., Chen, J., & Li, X. (2015). Alarm system optimization in process industries. \*Journal of Process Control, 35\*(2), 105–118.
2. Qian, F., et al. (2020). Detecting alarm floods with machine learning. IEEE Transactions on Industrial Informatics.
3. Lee, J., & Lee, Y. (2021). Predictive analytics for alarm prioritization. Computers & Chemical Engineering.
4. Ma, X., et al. (2018). Gradient boosting methods in predictive maintenance. Expert Systems with Applications.
5. Anderson, P., et al. (2022). Visual analytics in industrial automation. Information Visualization Journal.