**DETECTION OF MALICIOUS LOGS IN ENTERPRISE RESOURCE PLANNING SYSTEMS (ERPs) USING MACHINE LEARNING**

**Help Guide Documentation**

**Introduction**

The system logs from an HDFS ERP system are the subject of a thorough examination that we give in this report. The collection contains a variety of system events connected to transactions, system faults, and user logins. We aim to investigate and derive significant insights from this dataset to support decision-making, anomaly identification, and system optimization.

**Data Description**

The dataset comprises the following columns:

* Time: The timestamp of each system event.
* Pid: Process ID associated with the event.
* Level: The log level of the event.
* Component: The component or module within the ERP system.
* Content: The detailed content of the log event.
* EventId: An identifier for the type of event.
* EventTemplate: A template or pattern for log events.
* Classification: Categorization of events as "Malicious" or "Not Malicious."

The dataset contains a total of 104,815 entries.

The unique counts for each column are as follows.



The data shows that the events occurred in a single day.

A classification column was added based on the event template description

**Event mapping**

A classification column was added based on the event template description

"Receiving block <\*> src: /<\*> dest: /<\*>": "Not Malicious",

"BLOCK\* NameSystem.allocateBlock:<\*>": "Not Malicious",

"PacketResponder <\*> for block <\*> terminating": "Malicious",

"Received block <\*> of size <\*> from /<\*>": "Not Malicious",

"BLOCK\* NameSystem.addStoredBlock: blockMap updated: <\*> is added to <\*> size <\*>": "Not Malicious",

"Received block <\*> src: /<\*> dest: /<\*> of size <\*>": "Not Malicious",

"<\*>:Transmitted block <\*> to /<\*>": "Not Malicious",

"<\*> Starting thread to transfer block <\*> to <\*>": "Not Malicious",

"BLOCK\* ask <\*> to replicate <\*> to datanode(s) <\*>": "Malicious",

"<\*> Served block <\*> to /<\*>": "Not Malicious",

"Verification succeeded for <\*>": "Not Malicious",

"writeBlock <\*> received exception <\*>": "Not Malicious",

"PacketResponder <\*> <\*> Exception <\*>": "Malicious",

"Deleting block <\*> file <\*>": "Malicious",

"Receiving empty packet for block <\*>": "Malicious",

"Exception in receiveBlock for block <\*> <\*>": "Malicious",

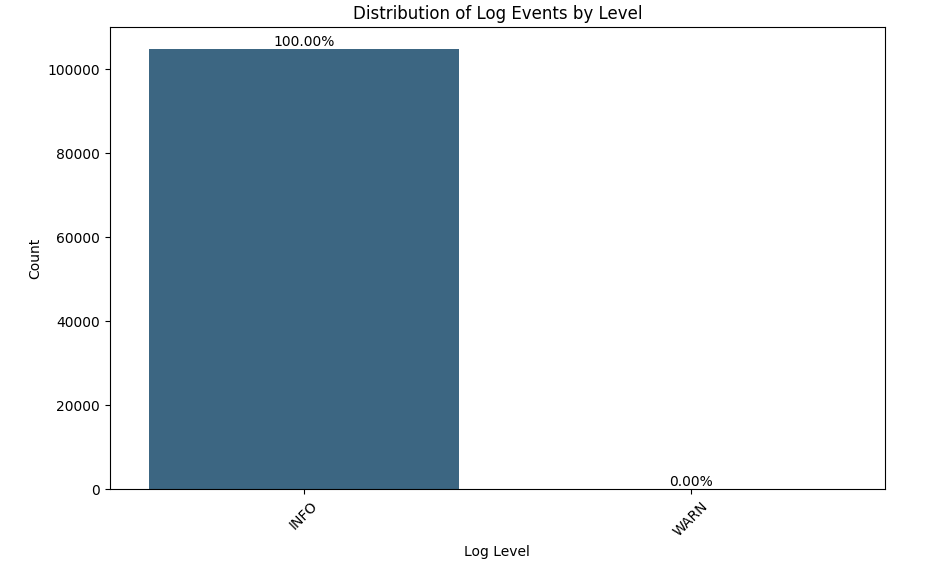
"BLOCK\* NameSystem.addStoredBlock: Redundant addStoredBlock request received for <\*> on <\*> size <\*>": "Not Malicious",

"PacketResponder <\*> for block <\*> Interrupted.": "Malicious",

"Changing block file offset of block <\*> from <\*> to <\*> meta-file offset to <\*>": "Malicious"

}

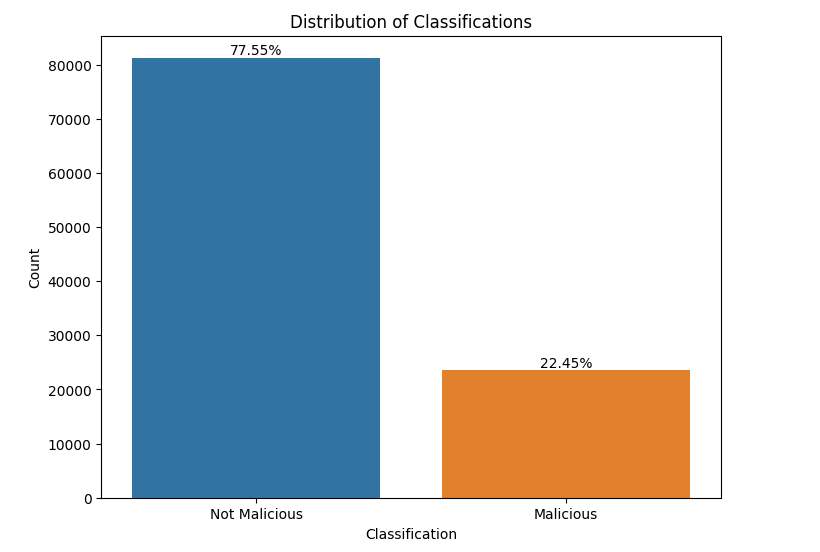
**Distribution of log events by level**



With a count of 104,814, or nearly all of the log events (99.99%), "INFO" is the most common classification for log events, according to the distribution of log events by "Level." There is only one log event that is labelled as "WARN," making up a tiny portion (0.01%) of all log events. This shows that "INFO" log events predominate the dataset significantly, but "WARN" log events are incredibly uncommon.

In summary, "WARN" log events are uncommon and comprise a tiny percentage of total log entries; the dataset is significantly biased towards "INFO" log events.

**Distribution of Classifications**

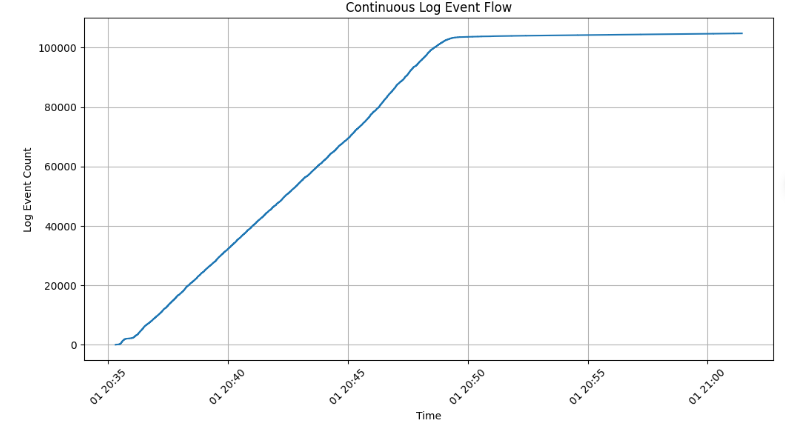


The distribution of log classifications was examined by examining the log events dataset, and two primary classes were identified: "Not Malicious" and "Malicious." Twenty-three thousand five hundred thirty-five log events classed as "Malicious," or roughly 22.45% of the dataset, and 81,280 log events classified as "Not Malicious," or approximately 77.55% of the total logs, make up the dataset.

The bulk of the log entries are classified as "Not Malicious," this distribution sheds light on how frequently log events occur within these two classes. Comprehending this distribution is crucial for developing efficient models for tasks linked to anomaly detection or security since it highlights the class imbalance that must be considered in subsequent analysis and model creation.

**Log event flow**

The cumulative log event flow is displayed on the graph. The graph, being a line graph, shows the total number of events that have transpired over the given period. The diagram indicates that there has been an increase in the number of incidents over time.



More specifically, the graph shows that the number of events that have occurred has increased from 0 to 100,000 in the period from 20:35 to 21:00. This means that there has been a steady increase in the number of events that have occurred over time because the system is experiencing increased load.

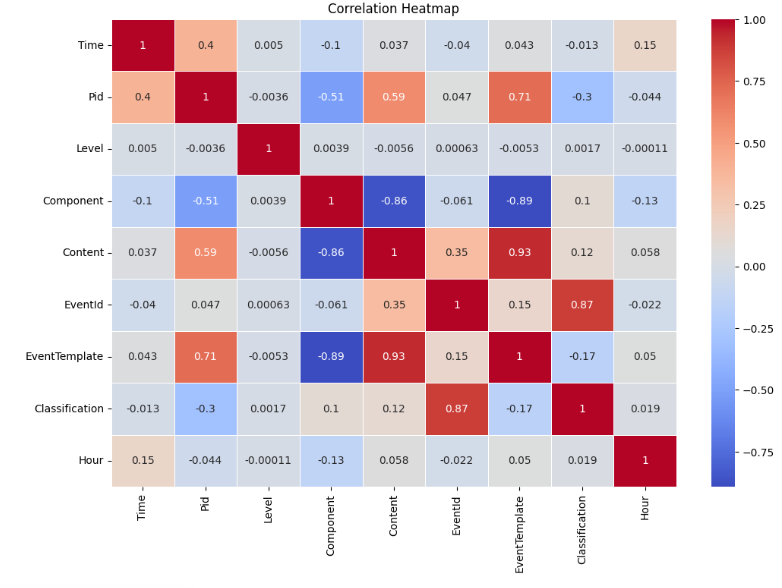
**Correlation Analysis**

Correlation is a statistical measure used to evaluate the strength and direction of a relationship between two variables. It quantifies how changes in one variable correspond to changes in another. Correlation can take values between -1 and 1, where:

* One indicates a perfect positive correlation: As one variable increases, the other increases proportionally.
* 0 indicates no correlation: The variables are independent of each other.
* -1 indicates a perfect negative correlation: As one variable increases, the other decreases proportionally.

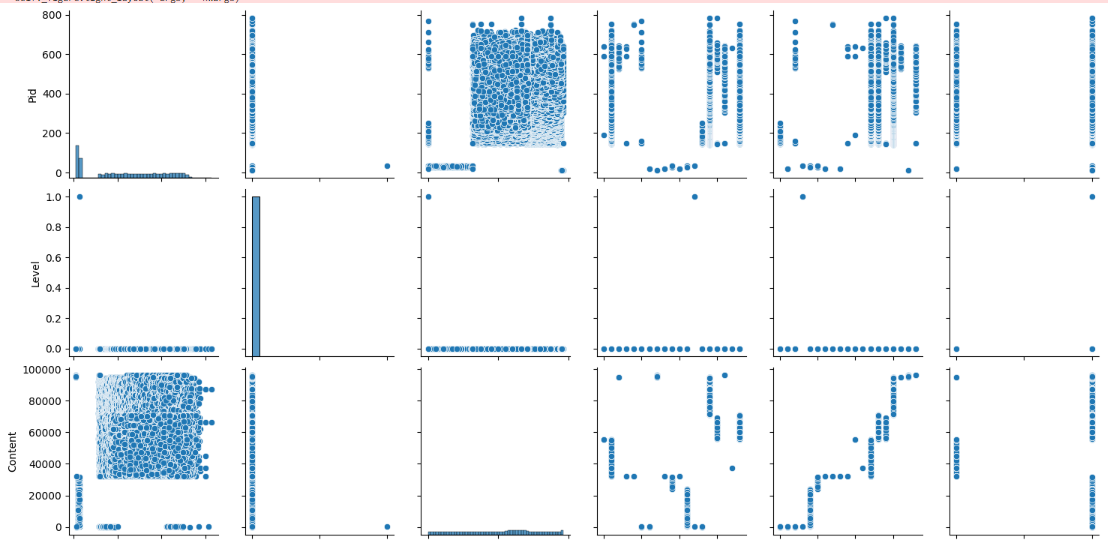
Correlation is often represented as a correlation coefficient, and the most commonly used coefficient is the Pearson correlation coefficient (Pearson's r), which measures the linear relationship between two variables.

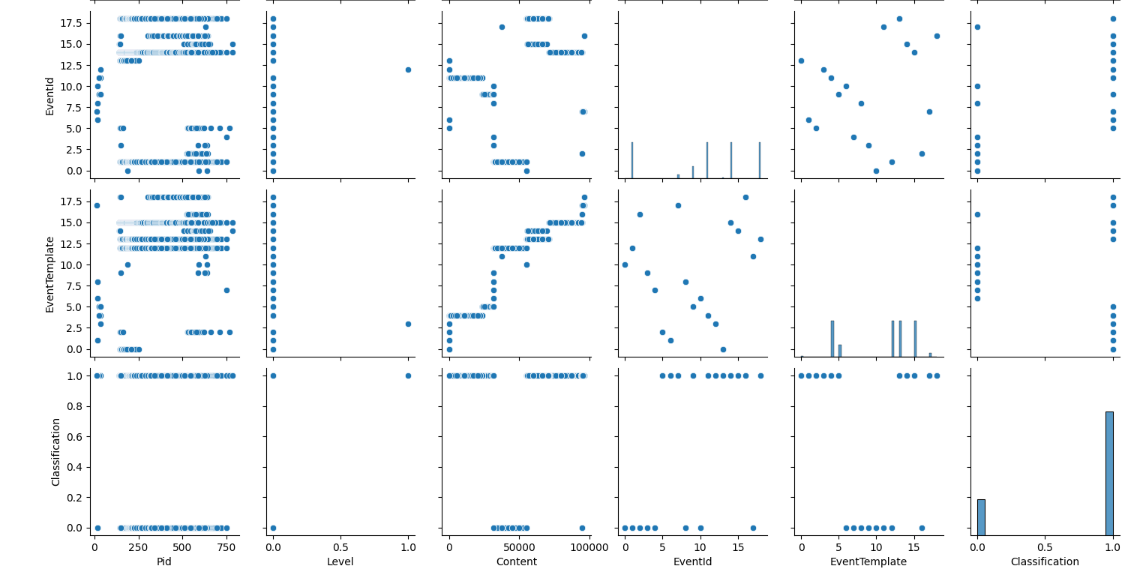
In the context of data analysis, correlation helps in understanding how two variables behave concerning each other.



1. Correlation between Pid and Component (Correlation: -0.512856):
   * The correlation between Process ID (Pid) and Component is moderately negative (-0.512856).
   * This negative correlation suggests that certain processes (Pids) are inversely related to specific system components. When a particular process is active (high PID), it may be associated with specific components.
2. Correlation between Pid and Content (Correlation: 0.589128):
   * The correlation between Pid and Content is moderately positive (0.589128).
   * This positive correlation implies that particular processes (Pids) are positively associated with specific content or events. Higher Pid values may correspond to more complex or higher-value content.
3. Correlation between Pid and Event Template (Correlation: 0.712707):
   * The correlation between Pid and Event Template exhibits a strong positive relationship (0.712707).
   * This suggests that specific processes (Pids) are strongly associated with particular Event Templates, and as Pid values increase, the Event Template values also increase significantly.
4. Correlation between Component and Content (Correlation: -0.858552):
   * The correlation between Component and Content is strongly negative (-0.858552).
   * This strong negative correlation indicates an inverse relationship.
   * It implies that different system components are associated with varying types or complexities of content.
5. Correlation between Component and Event Template (Correlation: -0.891893):
   * The correlation between Component and Event Template is strongly negative (-0.891893).
   * This strong negative correlation suggests that different components have a significant inverse relationship with specific Event Templates.
   * As Component values change, Event Template values exhibit a strong inverse correlation.
6. Correlation between Content and Event Template (Correlation: 0.926506):
   * The correlation between Content and Event Template is strongly positive (0.926506).
   * This positive correlation implies a robust positive relationship between the complexity or type of content and the associated Event Templates.
7. Correlation between Event Id and Classification (Correlation: 0.874682):
   * The correlation between Event ID and Classification is moderately positive (0.874682).
   * This suggests that certain Event IDs may be related to specific classifications.

**Scatter plot showing the relationship of each column to the other.**





The variables Content and Event Template still show a high correlation.

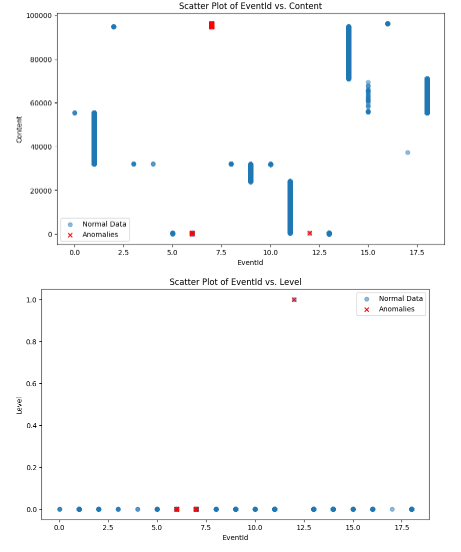
**Anomaly detection**

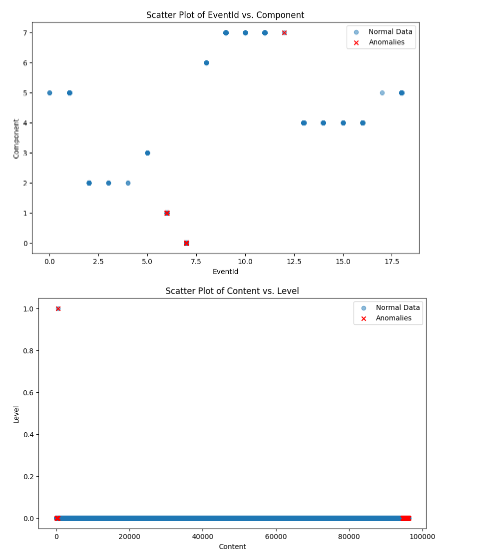
In the anomaly detection process, we employed both z-scores and dedicated anomaly detection models to effectively identify and visualize anomalies within the dataset. This multi-faceted approach ensured that anomalies were comprehensively addressed and provided a more robust anomaly detection process.

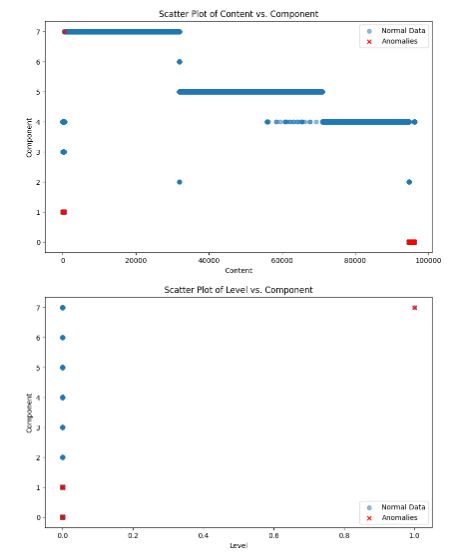
**Z-Scores:**

Z-scores are a statistical technique used to measure how many standard deviations a data point is from the mean. By calculating the z-scores for each numerical column, we were able to identify data points that significantly deviated from the column means. Data points with high absolute z-scores exceeding a predetermined threshold were flagged as anomalies. Z-scores served as an initial and fundamental step in our anomaly detection process, allowing us to detect outliers based on statistical deviations.

The following are the scatter plots showing the anomalies in red.







**Anomaly Detection Models**:

In addition to z-scores, we incorporated machine learning-based anomaly detection models. We used the Isolation Forest model and other anomaly detection techniques to enhance our ability to identify anomalies. These models employ advanced algorithms to isolate anomalies from most data points by leveraging characteristics such as isolation depths and decision trees.

By combining z-scores and machine learning-based models, we created a comprehensive approach to anomaly detection. This allowed us to capture anomalies that may not have been detected using a single method. The outcome was a more thorough identification and visualization of anomalies in the high-frequency structured log data.

This combined approach demonstrates the effectiveness of utilizing statistical and machine learning techniques to uncover anomalies, providing a robust framework for anomaly detection in diverse datasets.

The following are the scatter plots showing the anomalies in red.

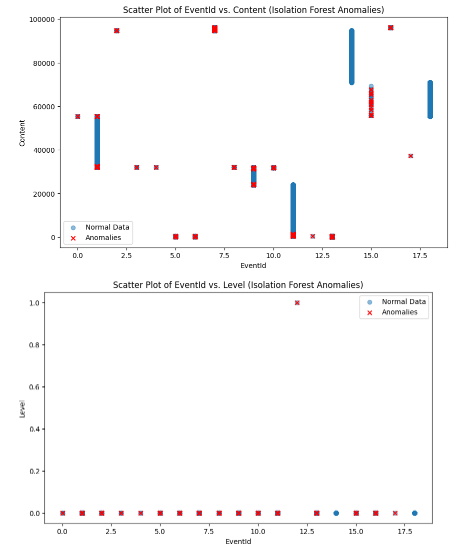
**Isolation Forest scatter plots**

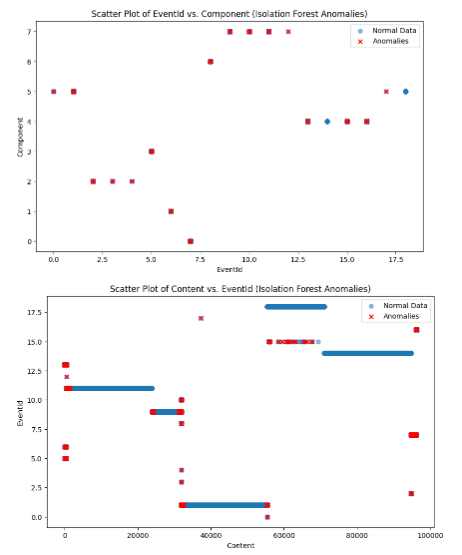
Method: Isolation Forest

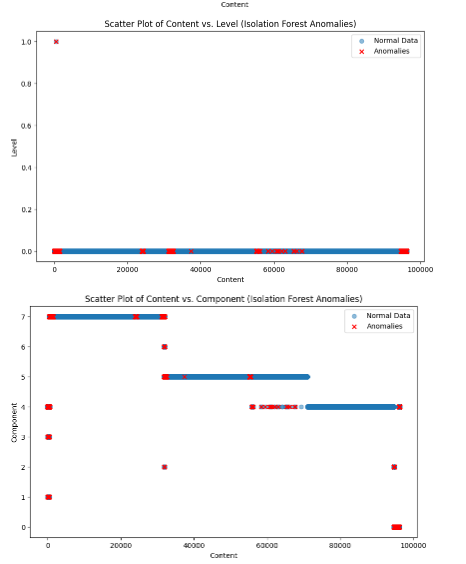
Number of anomalies: 5217

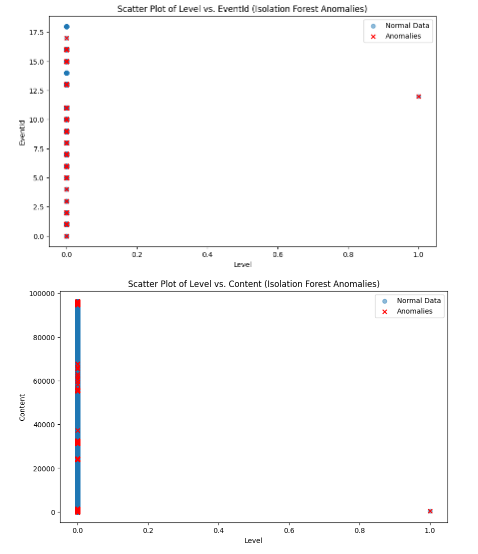
Number of average data points: 99598

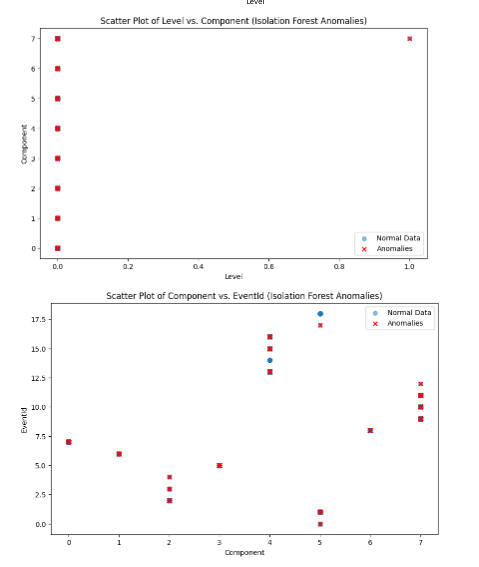
Percentage of anomalies: 4.98%

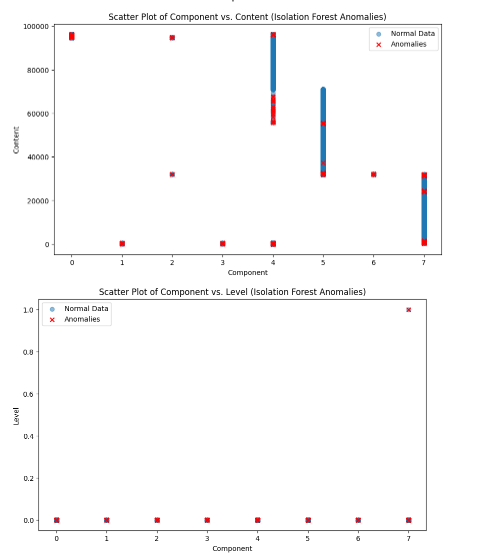












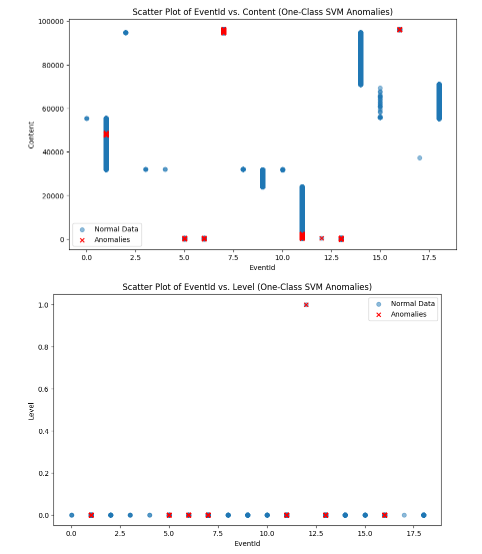
**One Class SVM**

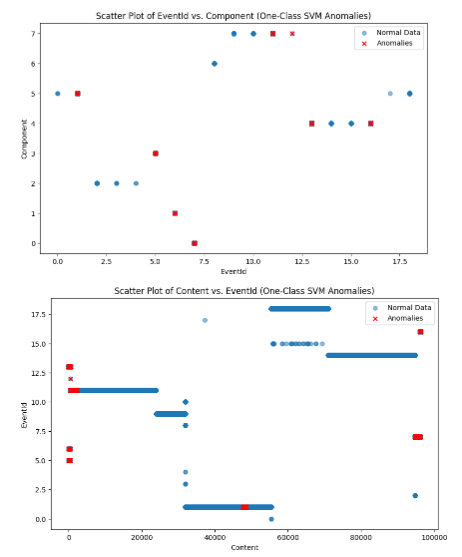
Method: One-Class SVM

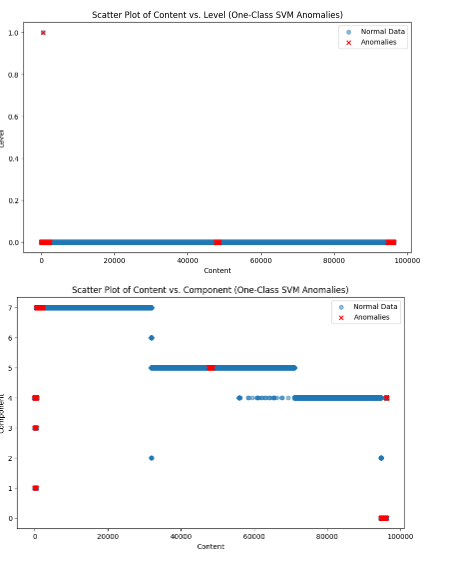
Number of anomalies: 5240

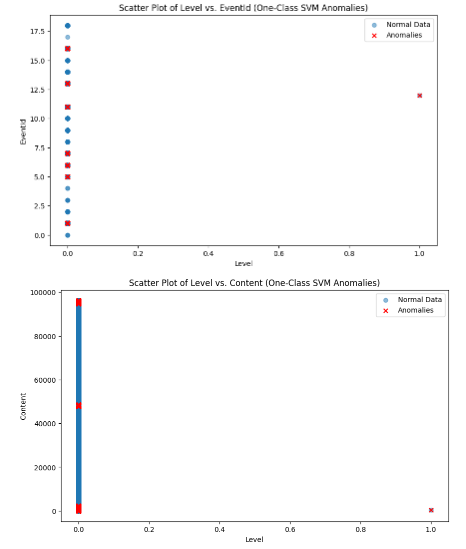
Number of average data points: 99575

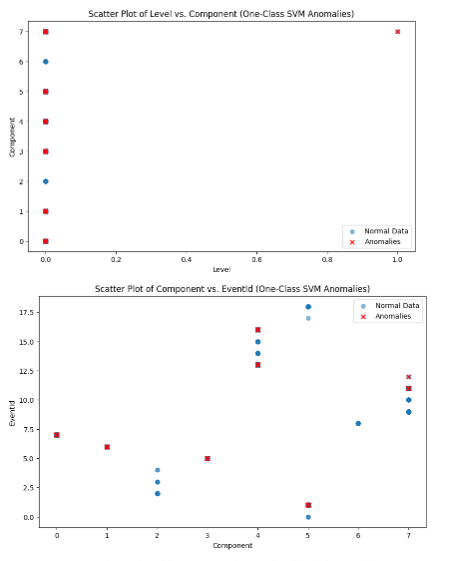
Percentage of anomalies: 5.00%

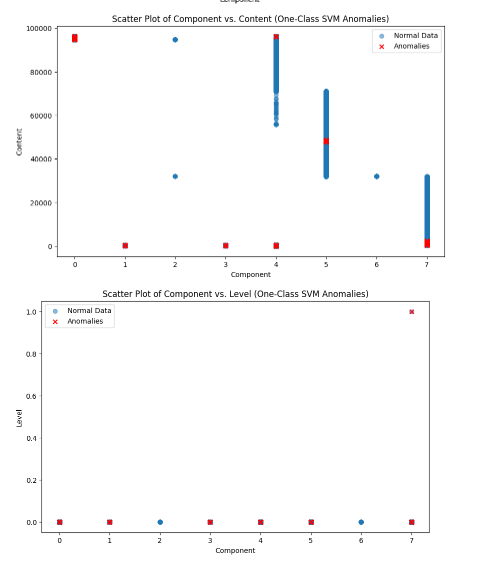




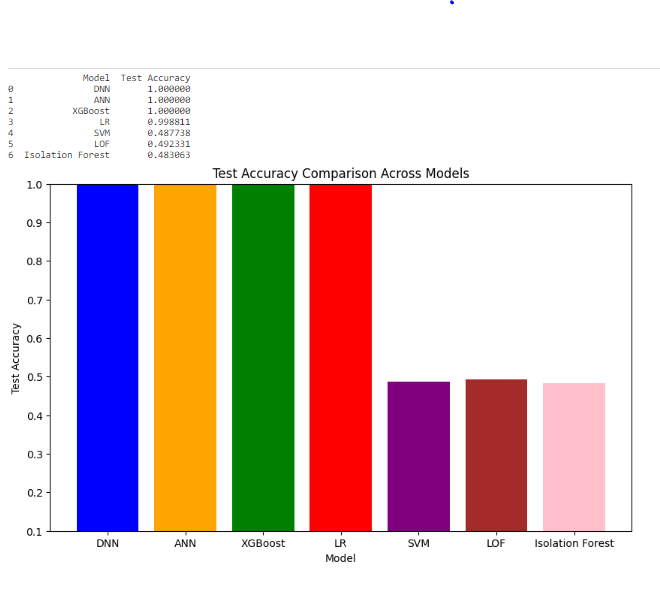






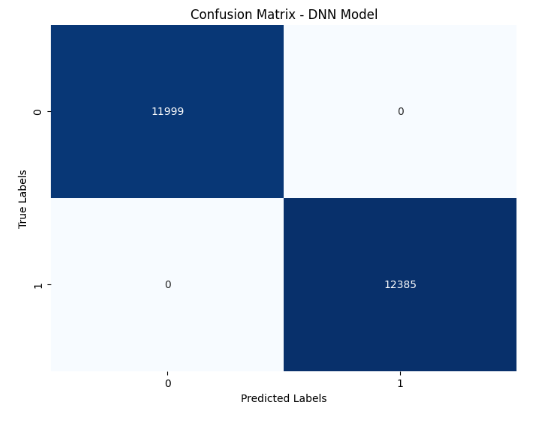


**MODEL DEVELOPMENT**

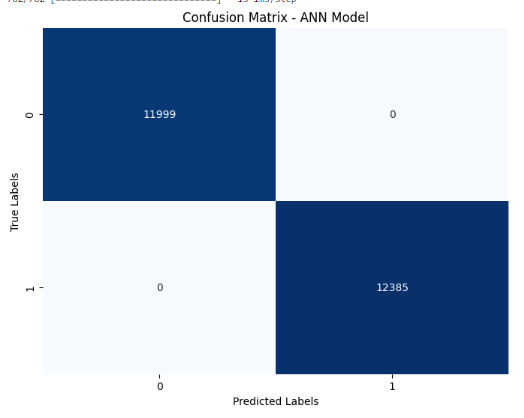


**MODEL DEVELOPMENT**

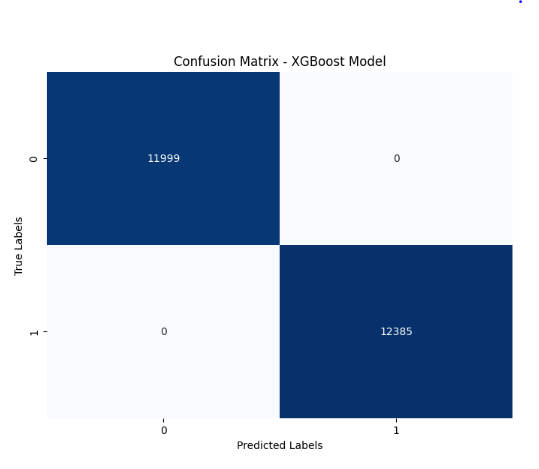
**Deep Neural Network (DNN):** The test accuracy of 1.000 was attained by both the DNN and ANN models, demonstrating their outstanding ability to discriminate between normal and aberrant occurrences. These deep learning models have successfully identified intricate patterns and characteristics in the data based on their high accuracy.



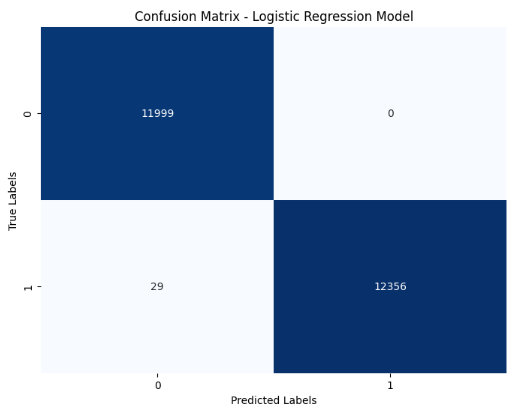
**Artificial Neural Net (ANN):** In line with the DNN's remarkable performance, the ANN model demonstrated a flawless test accuracy of 1.000. This further illustrates how neural networks can identify anomalies in structured log data.



**XGBoost:** This model's strong prediction performance was demonstrated with a test accuracy of 1.000. An excellent option for anomaly detection is the gradient-boosting algorithm XGBoost, which is renowned for its effectiveness and high precision.

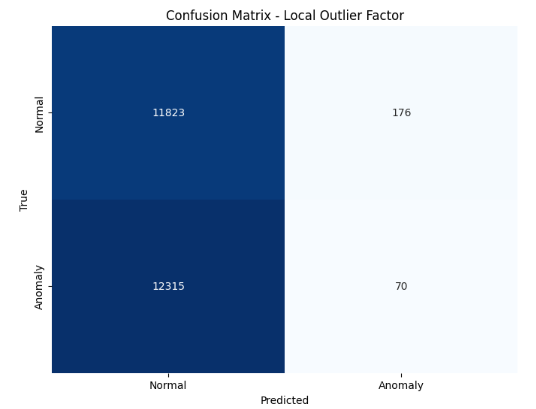


**Logistic Regression (LR):** Although the LR model yielded an exceptionally high test accuracy of 0.998811, XGBoost and the deep learning models significantly surpassed it. Still, it is a valuable method for finding anomalies, mainly when readability and ease of use are crucial.



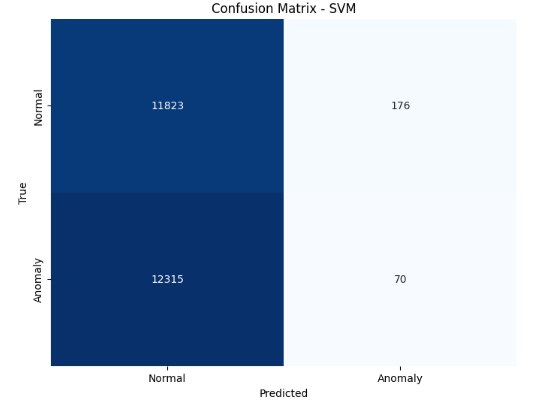
**Local Outlier Factor (LOF)**

The LOF model achieved an accuracy of 0.49233103674540685, indicating moderate success in identifying anomalies. The confusion matrix shows a relatively balanced performance, with positives and negatives.



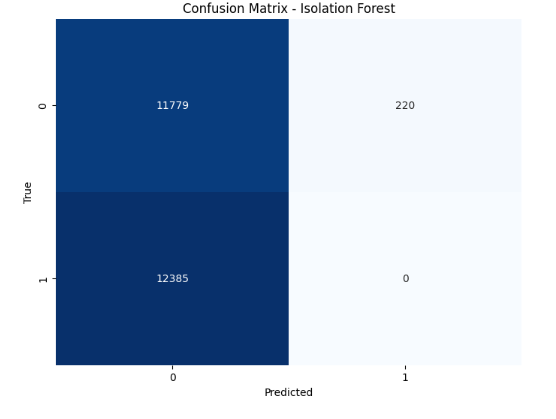
**Support Vector Machine (SVM)**

The SVM model attained an accuracy of 0.4877, indicating a reasonable ability to distinguish between normal and abnormal log events. The confusion matrix highlights the model's performance:

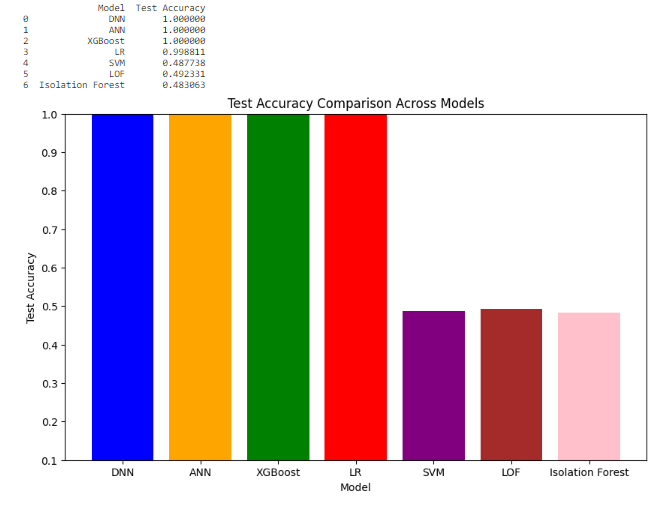


**Isolation Forest**

The Isolation Forest model achieved an accuracy of 0.483. The confusion matrix illustrates the model's performance:



**TEST ACCURACY COMPARISON ACROSS MODELS**



**CONCLUSION**

In conclusion, the comprehensive evaluation of various anomaly detection models has yielded promising results for their application in real-world Enterprise Resource Planning (ERP) environments. The deep learning architectures, including the Deep Neural Network (DNN) and Artificial Neural Net (ANN), showcased exceptional accuracy, achieving a perfect test accuracy 1.000. These models demonstrated a remarkable ability to discern intricate patterns and anomalies in structured log data.

The XGBoost model, a powerful gradient-boosting algorithm, also exhibited robust predictive performance with a test accuracy 1.000. Its effectiveness and precision in anomaly detection further underscore its reliability in identifying unusual events within the log data.

While Logistic Regression (LR) achieved a commendable test accuracy of 0.998811, it was slightly surpassed by the deep learning models and XGBoost. Nevertheless, LR remains a valuable option, particularly for scenarios where readability and ease of use are critical factors.

Ensemble methods and local outlier detection models, including Local Outlier Factor (LOF), Support Vector Machine (SVM), and Isolation Forest, provided additional insights into anomaly detection. LOF demonstrated moderate success, while SVM and Isolation Forest showed reasonable capabilities in distinguishing between normal and abnormal log events.

These findings suggest that the selected models can enhance ERP system monitoring and security. The high accuracy achieved by these models positions them as practical tools for early detection of security breaches or system issues, contributing to a more robust and secure ERP environment.