CMSC 25-342

Anomaly-Based Intrusion Detection for Networks

Preliminary Design Report

Margarette Duckett, Kareem Aboulhosn , Michael Ott , Samuel Dominguez Jacobo

Project Advisor

Nicholas J. Kaminski

Idaho National Laboratory

Faculty Advisor

Milos Manic

Executive Summary

The executive summary highlights the key points of the document. While your advisor(s) and sponsor are expected to read the document in detail, others may only read the summary looking for a brief overview of the report. Casual readers may look at the summary to decide if they would like to continue reading. Some, more senior decision makers (e.g. executives), may read the summary to help make decisions regarding the future of the project (e.g. continuation, financing, resource allocation, etc.). It is important that all readers get a complete sense of the project, including purpose, primary objectives, design requirements, deliverables, work done to date, and timeline, among other required components provided in a table of contents. Summaries should be considered as "stand-alone" containing a complete account of the essential points of the document in chronological order of the document. Particular focus should be placed on the first sentence in order to draw readers in and should explicitly include the "who, what, and why" of the project. The executive summary is usually between half a page and a full page.

Note: The Executive Summary should be updated between major reports as more knowledge is acquired and understanding of the project expands. For example, when submitting Preliminary Design Report in December 2024, make sure you update this page to reflect the progress on the project since the submission of Project Proposal in early October 2024.

Table of Contents

Section A. Problem Statement

Anomaly-based intrusion detection systems (IDS) are vital components in cybersecurity, identifying suspicious activities by recognizing deviations from normal behavior patterns within a network. While these systems are effective at detecting known threats, a significant challenge persists in their ability to detect zero-day attacks—cyber threats that exploit previously unknown vulnerabilities. Traditional IDS solutions, particularly signature-based detection methods, struggle to identify these novel attacks due to their reliance on pre-existing knowledge of attack signatures.(Wang & Su, 2019)

The key challenge lies in developing an IDS that not only excels at detecting known attacks but can also proactively identify zero-day attacks in real-time without relying on established signatures. Current anomaly-based IDS show promise, but they often suffer from high false-positive rates and difficulty distinguishing between benign and malicious anomalies, limiting their overall effectiveness. Additionally, many systems are trained on outdated datasets that fail to capture the rapid evolution of cyber threats and modern network environments. As the industry continues to expand and the nature of cyberattacks grows increasingly sophisticated, these static datasets hinder the ability of IDS to stay current, making it harder to detect emerging threats that exploit new vulnerabilities.

We propose enhancing anomaly-based IDS through the integration of machine learning techniques, particularly decision trees, combined with advanced statistical analysis. This approach could improve the system's ability to distinguish between benign anomalies and actual zero-day attacks by learning complex patterns in real-time network data. This hybrid approach could improve the detection of zero-day attacks by better understanding patterns in normal system behavior and flagging subtle deviations that signal potential novel threats. By leveraging decision trees for their interpretability and combining them with statistical anomaly detection methods, we can create a more robust, proactive intrusion detection system capable of evolving with emerging cybersecurity challenges. (Sharma & Suryakanthi, 2013)

Note: The problem statement should be updated between major reports as more knowledge is acquired and understanding of the project expands.

Section B. Engineering Design Requirements

This section describes the goals and objectives of the project, as well as all **realistic constraints** to which the design is bound. It is meant to provide a structure that helps to formulate the problem. Design requirements are often derived from client or stakeholder needs. They may consider benchmarking against or improving on currently available solutions, providing novel techniques or design solutions, integration with existing components, systems, or equipment, required codes and standards, general observations of the problem space, etc. Describe how the requirements provided below were researched and decided upon. Common design requirements often include considerations of the design efficacy, cost, safety, reliability, usability, and risk, among others.

Note: The design requirements should be revisited between major reports to ensure that the design objectives and constraints still accurately reflect the client's needs and project goals and to make sure that the team is on track to meet all goals and objectives.

Note: The codes and standards section is not required for the Project Proposal, but is required for all subsequent reports. This section should be comprehensive and thorough, requiring a significant research effort.

B.1 Project Goals (i.e. Client Needs)

The primary goal of this project is to develop an effective hybrid Intrusion Detection System (IDS) for wireless 5G networks. The IDS will be based on a combination of decision tree algorithms and the Naive Bayes model, designed to detect zero-day vulnerabilities while minimizing false positive rates. The following are the specific goals of the project from the client's perspective:

- **Develop a hybrid IDS** using decision tree and Naive Bayes models for anomaly-based detection in 5G networks.
- **Train the IDS** on historical datasets to build a robust model capable of identifying potential security breaches.
- **Test the IDS** on more recent datasets to evaluate its effectiveness, particularly in detecting zero-day vulnerabilities.
- **Reduce false positive detection rates** to ensure accurate and reliable threat detection.

- **Measure system performance** in terms of detection accuracy, efficiency, and scalability within a 5G wireless network environment.

B.2 Design Objectives

The design objectives for this Naive Bayes hybrid with Decision Trees anomaly-based intrusion detection system prioritize **high detection accuracy** by minimizing false positives and negatives, while also ensuring **scalability** to handle the large data volumes typical of 5G networks. The system will be **adaptable to zero-day attacks**, using machine learning to detect unknown threats, and **robust against evasion techniques** like traffic encryption and obfuscation. Additionally, it will adhere to **data privacy and security standards** such as NIST and ISO/IEC 27001, ensuring encrypted transmission of sensitive data and compliance with industry regulations.

- **High Detection Accuracy**: The system should accurately detect and classify intrusions, minimizing false positives and negatives. This includes leveraging the hybrid model (Naive Bayes and Decision Trees) to enhance the precision of detecting anomalies in 5G network traffic.
- **Adaptability to Zero-Day Attacks**: The system should be designed to detect zero-day attacks, using machine learning techniques to recognize patterns and anomalies that do not fit within predefined attack signatures.
- **Scalability**: The design must be scalable to handle the large volumes of data generated by 5G networks. The system should be able to efficiently process real-time data, adapting to increasing network size and complexity.
- **Robustness Against Evasion Techniques**: The system should be designed to withstand common evasion techniques used by attackers, such as packet fragmentation, encryption, and traffic obfuscation, ensuring comprehensive protection
- **Data Privacy and Security**: Ensure that the system adheres to data security best practices, encrypting sensitive data during transmission and complying with relevant cybersecurity standards (e.g., NIST, ISO/IEC 27001).

B.3 Design Specifications and Constraints

A list of design specifications and constraints include all limitations, restrictions, and requirements of the design. They are firm limits that must be met for a design to be acceptable and are ultimately used to measure the success of a design. Each specification or constraint should map to one or more design objective(s) and explicitly state *how the design* will meet the

objectives. Specifications and constraints should be specific and are often numerical. They must be measurable or testable to prove that the design has met all of the design objectives. Numerical metrics may include qualifying statements such as “at least,” “at most,” “between,” “exactly” or include a set of discrete values. Avoid subjective, untestable constraints (e.g. “environmentally friendly”, “user friendly”, “nice looking”, etc.).

Realistic constraints can come take on a variety of forms including accessibility, aesthetics, codes, constructability, cost, ergonomics, extensibility, functionality, interoperability, legal considerations, maintainability, manufacturability, marketability, policy, regulations, schedule, standards, sustainability, or usability. Examples of physical constraints might include numerical limits or ranges on overall size envelope, weight, pressures, stresses, flow rates, voltages, current, power consumption, hardware limitations, data constrains, interoperability, etc. Other constraints might include production unit cost, expected part/device life, or maintenance requirements.

- Design must identify at least 70% of zero-day threats in testing scenarios, which are previously unseen by the system during training.
- Design should operate normally within industry benchmarks or existing IDS solutions (e.g., aiming for a lower FPR than traditional rule-based systems, which may have FPRs ranging from 10% to 30%).
- Design should accurately detect real threats at least 95% of the time, ensuring it captures critical incidents while reducing the impact of false positives.
- Design will use metrics to ensure a good balance between minimizing false positives and maintaining high detection rates.
- Design must identify at least 70% of zero-day threats in testing scenarios, which are previously unseen by the system during training.

Section C. Scope of Work

The project scope defines the boundaries of the project encompassing the key objectives, timeline, milestones and deliverables. It clearly defines the responsibility of the team and the process by which the proposed work will be verified and approved. A clear scope helps to facilitate understanding of the project, reduce ambiguities and risk, and manage expectations. In addition to stating the responsibilities of the team, it should also explicitly state those tasks which fall *outside* of the team’s responsibilities. *Explicit bounds* on the project timeline, available funds, and promised deliverables should be clearly stated. These boundaries help to avoid *scope creep*, or changes to the scope of the project without any control. This section also defines the

project approach, the development methodology used in developing the solution, such as waterfall or agile (shall be chosen in concert with the faculty advisor and/or project sponsor). Good communication with the project sponsor and faculty advisor is the most effective way to stay within scope and make sure all objectives and deliverables are met on time and on budget.

C.1 Deliverables

The project deliverables are those things that the project team is responsible for providing to the project sponsor. They are the things that are to be produced or provided as a result of the engineering design process. Some deliverables might include a specific number of alternative designs, required analyses to prove the design meets specifications, detailed machine drawings, functional diagrams or schematics, required computer code, flow charts, user manuals, desktop models, and functioning prototypes. A design "proof of concept" is not specific and should be more clearly defined. Academic deliverables include the team contract, project proposal, preliminary design report, fall poster and presentation, final design report, and Capstone EXPO poster and presentation. Provide a bulleted list of all agreed upon project deliverables.

In order to mitigate risks associated with the completion and delivery of the project deliverables, provide an outline of the most potentially disruptive, foreseeable obstacles. Some important issues to discuss with the design team, sponsor, and faculty advisor include the following:

- Team Contract
- Project Proposal
- Design Report
- Fall poster/Presentation
- Final design report
- Capstone EXPO poster/presentation
- Psuedocode/Code
- Test Summary Report
- Working Prototype
-

C.2 Milestones

Milestones are major project phases or tasks that need to be completed in order to ensure the project deliverables. They may include, among other things, completion of calculations, the development of a computational model, completion of an analysis, set-up of an experiment,

completion of data acquisition, purchasing of hardware, assembly of a prototype, completion of testing procedures, development of required code, completion of wiring, post processing, etc.

A good rule of thumb is to break the project down into tasks of no larger than 2-3 weeks in length. These can be individual or group tasks. Breaking down the project into tasks/milestones gives the team and the advisor/sponsor a realistic understanding of what can be done in the allotted time. In an agile development approach, later tasks are expected to be adjusted (or changed) as the team works with the earlier developed tasks.

The amount of time it will take to accomplish each milestone and the approximate date that each milestone will be completed should be considered. Do not underestimate the time that it takes to write and prepare major reports and presentation materials. All deliverables and milestones should be included in the project timeline found in Appendix 1. Provide a summary table of all project milestones including required times and completion dates here.

Literature Review

Research existing systems and methodologies in IDS.

2 - 3 weeks

Requirements Gathering

Complete the project proposal, outlining the trajectory of our project

1 - 2 weeks

Dataset Collection & Preprocessing

Gather and clean datasets for the training of the AIDS

2 weeks

Model Development

Develop decision tree model and statistical analysis methods

3 - 6 weeks

Prototype Testing

Using the clean data conduct in-depth testing

1 - 2 weeks

Analysis of Results

Draw conclusions from testing and tweak the model if needed

1 - 2 weeks

Upgrades to the model

From the conclusions implement updates as needed

2 - 4 weeks

Presentation Preparation

From the conclusions develop a poster presenting the findings

1 week

Note: While the project scope, deliverable, and milestones are not intended to change throughout the project, this section should be revisited between major reports to ensure that it still accurately reflects the expectations and requirements of the project team, client, and faculty advisor. Any changes to the project scope, deliverable, and milestones should be thoroughly discussed and mutually agreed upon by all parties. Any changes to this section should be documented and justified in detail

C.3 Resources

Resources needed for project completion should be listed at the proposal stage. These resources can either be purchased within the Project Budget, or provided by the project sponsor. Some examples are: hardware such as HPCs or servers, software such as IDEs, data analysis platforms or version control systems. Access to cloud computing services may also be necessary to scale certain procedures. Additionally, databases containing operational data for testing, as well as libraries or APIs relevant to predictive analytics and machine learning may be required.

Section D. Concept Generation

Concept 1: Hybrid IDS Using Decision Trees and Naive Bayes

- Combine decision trees for their interpretability and Naive Bayes for probabilistic anomaly detection to classify network traffic. Decision trees will capture distinct feature patterns, while Naive Bayes will evaluate probabilities based on network behavior trends

- Pros
    - High interpretability
    - Low computational cost
    - Enhanced detection of complex patterns
- Cons
    - It may require careful tuning
    - Could generate false positives
- Risk of failure: Inaccurate feature engineering or reliance on outdated datasets may reduce accuracy

Concept 2: Deep Learning-Based Anomaly Detection

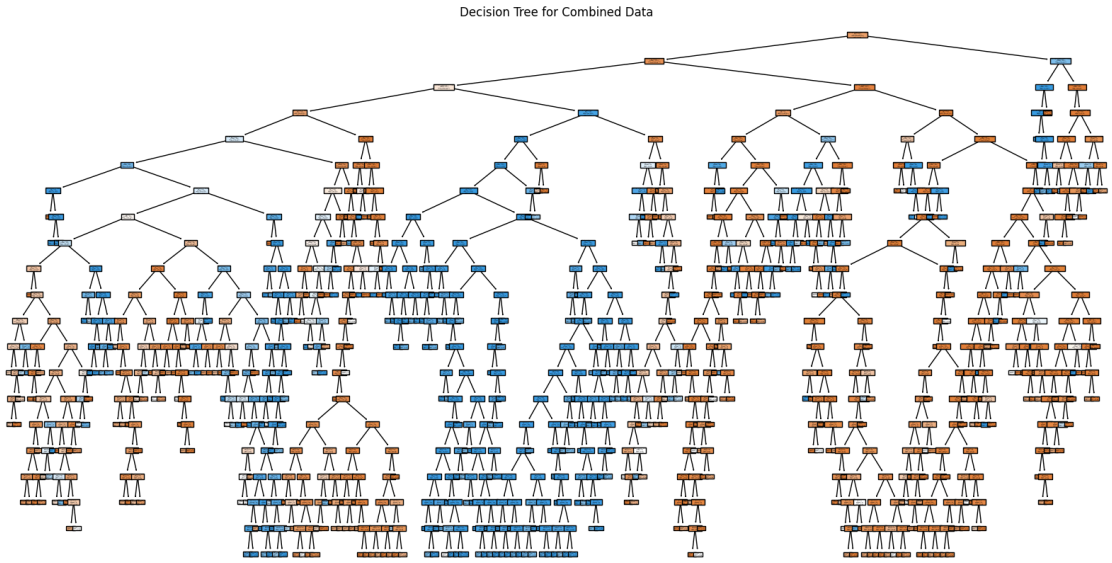
- Use a neural network trained on the CICIDS2017 dataset to learn representations of normal traffic patterns and identify deviations as anomalies
- Pros
    - Handles complex patterns
    - Adaptive and scalable for large datasets
- Cons
    - Requires significant computational resources
    - Difficult to interpret model decisions
- Risk of failure: Overfitting during training or underperformance on unseen data due to insufficient generalization

Concept 3: Statistical and Rule-Based Hybrid IDS

- Integrate statistical analysis with a rule-based engine for identifying zero-day threats. Statistical methods detect outliers, while rules provide domain-specific insights.
- Pros
    - Simple and efficient
    - Low resource requirements
- Cons
    - Limited scalability
    - Rule creating requires manual intervention
- Risk of failure: Rules may fail to adapt to new attack vectors, and statistical methods may struggle with subtle anomalies

Additional Components

- Feature Selection: Evaluate feature importance for improved model performance
- Data Preprocessing: Ensure proper handling of imbalanced data
- Data Visualization: Incorporate real-time dashboards to visualize traffic patterns and anomalies



Section E. Concept Evaluation and Selection

Selection Criteria

1. Performance: Accuracy and precision in detecting anomalies
2. Cost: Computational and time resources required
3. Scalability: Ability to adapt to larger datasets or increased traffic
4. Reliability: Consistency of detection under varied conditions
5. Interpretability: Ease of understanding and debugging model decisions

Criteria	Weight	Hybrid IDS	Deep Learning	Statistical IDS
Performance	0.4	8 (3.2)	9 (3.6)	6 (2.4)
Cost	0.2	9 (1.8)	6 (1.2)	8 (1.6)
Scalability	0.15	8 (1.2)	9 (1.35)	6 (0.9)
Reliability	0.15	8 (1.2)	8 (1.2)	7 (1.05)
Interpretability	0.1	9 (0.9)	6 (0.6)	7 (0.7)
Total Score		8.3	8.0	6.65

Selection Rationale: The hybrid IDS scores highest overall due to its balance of interpretability, performance, and cost-efficiency.

Section F. Design Methodology

F.1 Computational Methods

- Use Python and libraries such as Scikit-learn for decision trees and Naive Bayes implementation.
- Evaluate model performance using metrics like accuracy, recall, F1-score, and precision.
- Use cross-validation to ensure generalizability.

F.2 Experimental Methods

- **Test Environment:** Simulate a network using the CICIDS2017 dataset with live attack scenarios if feasible.
- **Tools:** Wireshark for traffic capture, and preprocessed Kaggle data for training.
- **Setup:** Split data into training (70%) and testing (30%) sets. Use synthetic data to simulate zero-day attacks.
- **Evaluation Metrics:** Track false positives, false negatives, detection latency, and resource usage.

F.3 Architecture/High-Level Design

- **System Components:**
    - Preprocessing Module: Cleans data and extracts features.
    - Machine Learning Module: Performs classification using the hybrid model.
    - Alerting and Logging System: Flags anomalies for review.
- **Pipeline Diagram:** Include stages from data ingestion to anomaly detection and alert generation.

F.5 Validation Procedure

- **Plan:** Validate the final model with:
    - The CICIDS2017 dataset for known attacks.
    - Simulated zero-day attack data for anomaly detection.
- **Timeline:** Prototype testing by mid-March; client feedback session in early April.
- **Feedback Collection:** Use structured surveys and observation notes during prototype demonstrations to capture client and stakeholder insights.

Section G. Results and Design Details

G.1 Modeling Results

We planned to utilize the previous group's algorithms to develop both of our components. Based on their work, we developed the code and tested each component using a fraction of the CICIDS2017 dataset to ensure proper functionality. Our primary goals were for each component to independently:

- Identify at least 70% of zero-day threats
- Detect real threats with at least 85% accuracy

Initial testing showed that the components could achieve these performance targets. This validation step was crucial before moving on to more comprehensive testing and integration phases.

G.2 Experimental Results

Each component was tested against the CICIDS2017 dataset, which includes both benign and malicious traffic. The experimental setup involved running these components through various partial datasets to evaluate their performance. We tracked metrics such as accuracy, precision, and recall, providing us with a comprehensive view of each component's effectiveness. Experimental results confirmed the components' ability to accurately detect and classify different types of network traffic. Both components exceeded our marker of 85% accuracy, providing the team with confidence to move forward with a hybrid approach.

The following image shows the results table for our Decision Tree component. The algorithm was tested on a fraction of the dataset, as previously mentioned, to ensure it was functioning correctly. The Decision Tree model achieved 99% accuracy, indicating it correctly classified the data provided. Additionally, it demonstrated a 99% precision rate, reflecting its ability to limit false positives, and a 98% recall rate, which measures the rate of correctly identified true positives while minimizing false negatives.

	Decision Trees
Accuracy	.9946
Precision	.9917
Recall	.9809
ROC AUC	.9982

The following image shows the results table for our Naive Bayes component. The algorithm was tested on a fraction of the dataset, as previously mentioned, to ensure it was functioning correctly. The Naive Bayes model achieved 96% accuracy, indicating it correctly classified the data provided. Additionally, it demonstrated a 96% precision rate, reflecting its ability to limit false positives, and a 96% recall rate, which measures the rate of correctly identified true positives while minimizing false negatives.

	Naïve Bayes
Accuracy	.9644
Precision	.9665
Recall	.9644

G.3 Next steps

After developing and validating individual components, we plan to integrate them into a cohesive hybrid IDS solution. Each prototype will be tested in stages to ensure seamless integration. Our testing will focus on real-world scenarios, utilizing the labeled data from the CICIDS2017 dataset. We will gather detailed performance metrics, including classification reports, to ensure our IDS meets the required standards.

As we advance with the hybrid approach, we will meticulously track performance changes and evaluate the overall effectiveness of our solution. This analysis will also consider the IDS's scalability and potential for implementation in industry settings. By continuously monitoring and refining our system, we aim to deliver a robust and efficient IDS solution capable of addressing the evolving challenges in cybersecurity.

G.4 Final Design Details/Specifications

The final IDS design will integrate the validated components into a single, efficient system. It will meet all design objectives and constraints, as demonstrated by our comprehensive testing. Specifications include:

- Design must identify at least 70% of zero-day threats in testing scenarios, which are previously unseen by the system during training.
- Design should operate normally within industry benchmarks or existing IDS solutions (e.g., aiming for a lower FPR than traditional rule-based systems, which may have FPRs ranging from 10% to 30%).
- Design should accurately detect real threats at least 95% of the time, ensuring it captures critical incidents while reducing the impact of false positives.
- Design will use metrics to ensure a good balance between minimizing false positives and maintaining high detection rates.
- Design must identify at least 70% of zero-day threats in testing scenarios, which are previously unseen by the system during training.

The thorough testing and validation will confirm that our design functions as intended, meeting all predefined objectives and constraints. This structured approach ensures that we deliver a robust, effective IDS solution within the given timeframe and resource constraints.

Section H. Societal Impacts of Design

Public Health, Safety, and Welfare

- **Safety**: An IDS helps protect critical infrastructure, personal data, and financial systems from cyber-attacks, which can have significant implications for public safety and welfare.
- **Health**: By securing healthcare systems, an IDS ensures that patient data remains confidential and that medical devices are not compromised, which is crucial for patient safety.

Societal Impacts

- **Interactions**: Providing the comfort of security for technology users can impact their willingness to interact. As technology grows the uncertainty of risk grows with it by securing sensitive information we encourage the acceptance of growth.

Political/Regulatory Impacts

- **Compliance**: The design must adhere to regulatory requirements such as GDPR, HIPAA, and other data protection laws, ensuring that the system is legally compliant.
- **Policy**: Successful deployment of IDS can influence cybersecurity policies and encourage governments to adopt stricter security measures.

Economic Impacts

- **Market Shift**: Advancements in the current IDS systems can disrupt the market by setting new standards of performance, forcing competitors to grow or lose a leading spot.
- **Costs**: Implementing IDS can lead to significant cost savings by preventing data breaches and minimizing financial losses associated with cyber-attacks.

Environmental Impacts

- **Energy Consumption**: As with most technology an increase in use that is not optimal can lead to a more significant carbon footprint.

Global Impacts

- **Trade**: As cybersecurity is a global concern, an effective IDS can enhance international trade by providing secure digital environments, fostering trust between trading partners.
- **Collaboration**: Global collaboration on cybersecurity standards can lead to more robust and universally accepted security practices.

Ethical Impacts

- **Privacy**: Ensuring that the IDS respects user privacy and operates transparently is crucial for maintaining trust.
- **Bias**: The design should be monitored for bias that it may unknowingly develop against a certain type of use case.

Unintended Consequences

- **False Positives/Negatives**: While aiming to reduce false positives and negatives, it's important to consider how these errors might affect users and the overall security posture.

Section I. Cost Analysis

We are fortunate to not have incurred cost during this process. With the use of open source data as well as free software we have been able to avoid cost thus far. If we were to scale this into a commercial product the following factors would drive up cost:

Development cost

- Initial setup would depend on the complexity of the system
- Maintenance would vary by customer and encompass annual cost

Operational cost

- Hardware to host and maintain the IDS cost will vary by requirements
- Licensing fees

Deployment cost

- Implementation will vary by size of the customer and the approach used
- Training either on site staff or remote for troubleshooting

J. Conclusions and Recommendations

Story of Our Design Process

We began by researching the current state of IDS technology, identifying industry struggles, and exploring innovative approaches to address these issues. We then decided on a hybrid approach, combining multiple IDS components for enhanced performance.

Once we decided on the hybrid approach and identified the components we would use as decision trees and naive bayes. We then focused on finding a suitable dataset and discovered a pre-processed version of the CICIDS2017 dataset. This dataset was instrumental in allowing us to focus on algorithm development and testing rather than data cleaning.

We were fortunate to have the previous group's algorithms for the components listed above. These algorithms were adapted to develop our independent IDS components. In saving time on the development of the components we will be able to further focus on the validity of our hybrid approach. The testing phase of the components involved validating our components against key metrics such as accuracy, precision, recall, and F1-score.

Lessons Learned

Our biggest takeaway this far has been the importance of research. Through our research we have been able to cut down valuable use of time. The research also helped us gain a strong foundation on the IDS and focus our efforts on a lacking part of the industry.

Future Advancements and Improvements

The individual components demonstrated high accuracy and reliability, with metrics indicating robust performance in detecting and classifying network traffic. This success has given us confidence as we move forward with the integration of these components into a hybrid IDS solution.

Looking ahead, the hybrid approach will be tested and refined. Our immediate next steps involve starting the development of the hybrid system post-break, with a focus on tracking performance changes and evaluating scalability for industry use.

Continuation and Future Work

We suggest that future teams focus on the following:

- Finalizing the hybrid IDS and conducting extensive testing
- Documenting all code, drawings, and results in detail
- Exploring the combination of other components

REFERENCES

1. [1]Wang, Z., & Su, X. (2019). *Anomaly detection based on deep learning: A survey*. Complex & Intelligent Systems, 5(4), 1-26. https://doi.org/10.1186/s42400-019-0038-7
2. [2]Sharma, N., & Suryakanthi, V. (2013). *Anomaly based network intrusion detection: A review*. 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, 1374-1379. https://doi.org/10.1109/ICACCI.2013.6637431
3. [3] Kaggle. (2021). CICIDS2017 Full Dataset. Retrieved from https://www.kaggle.com/datasets/sweety18/cicids2017-full-dataset