Journal of Information Systems Engineering and Management
2025,10(5s)
e-ISSN:2468-4376
https://www.jisem-journal.com/

Research Article

Accelerating Intrusion Detection Dataset Analysis- A Framework Using AutoGen Agents for CIC-IDS 2017

Prof. Nitin W. Wanhade, Dr. Sagar V. Joshi, Dr. Saurabh Saoji , Prof. Sarika N. Patil, Prof. Sushma Bhsole
Nutan Maharashra Institute of Engineering & Technology, Talegaon Dabhade, Pune, India
Corresponding author: Nitin W. Wankhade, nitin.wankhade@gmail.com

ARTICLEINFO

Received: 09 Oct 2024

Revised: 10 Dec 2024

Accepted: 21 Dec 2024

ABSTRACT

An IDS is a vital component in securing any network, however, the practical operation of an IDC is often dependent upon reasonable response times for the data with a huge volume. In this paper, we attempt to enhance the analysis of the CIC-IDS 2017 dataset using AutoGen, a deep learning model framework related to state-of-the-art. AutoGen performs a lot of the work automatically without requiring human intervention bottlenecks such as data preprocessing, feature engineering, or even model training thus saving a lot of time and work when developing an IDS. We compared the performance of AutoGen against prompt-based language models by focusing on task completion metrics along with three additional metrics: Humane Evaluation score, time taken, and resource overhead. The results exhibited that AutoGen is far superior to conventional ones in every way possible. In summary, the findings of this study demonstrate AutoGen's popularity for the future of intrusion detection through its data analysis function in the bias of the entire system performance parameter.

Categories: AI applications, Machine Learning (ML), Machine Learning (ML)

Keywords: autogen, data analysis, cic-ids 2017 dataset, generative pre-trained trans-former, automation, model training.

INTRODUCTION

In the ever-changing domain of computer security, Intrusion Detection Systems (IDS) have become ubiquitous in protecting virtual environments against a variety of attacks [1]. These threats keep getting more advanced and abuse the system requiring that threats be handled by threats from well-equipped systems. More often than not, traditional methods of intrusion detection systems (IDSs) tend to not fit well into the larger processing of large amounts of data which may in turn hinder their efficiency in countering these threats. However, new technologies such as artificial intelligence (AI) and machine learning (ML), developed in the last decade, help to address these issues [2].

This work presents an investigation concerning the potential application of AutoGen for the analysis of intrusion detection datasets based on the example of the CIC-IDS 2017 dataset [3]. AutoGen performance is evaluated relative to that of conventional prompt-based language models using several key parameters: Task Completion Rate, Human Evaluation score, Processing Time, and Computational Resources [4].One such example is that of AutoGen, which is a highly generative pre-trained transformer equipped with predictive language capabilities and many other features [1]. In addition to its linguistic features, AutoGen also helps improve the processes of important steps involved in the operation of IDS such as data preprocessing, feature engineering, and model training, all of which are highly manual [2].

As a result, the present research agrees with the findings reported by some scholars on their subject who stated that depicting or using AutoGen in place of the traditional model information retrieval systems is advantageous [3]. The findings of this study also help in appreciating the emerging picture of AutoGen Technology in the context of IDS systems and its datasets. In particular, due to the new functionality of the systems, organizations can greatly improve their cybersecurity systems and thus offer more efficient and transversal protective measures against increasing levels of cyber-attacks [4]. Intrusion Detection System datasets contain several problems such as largeness, complexity, class imbalance, and so on as well as the rapid development of fresh menaces. In most cases, visual comprehension involves a lot of work and is futile where many data points exist, and complex feature interactions play a role. At the same time, fresh attack vectors bring challenges concerning new techniques to fight old problems.

AutoGen solves these problems by integrating data cleaning, feature selection, model training, and model validation auto, thus increasing productivity [1]. Studies show that its scalability helps to deal with large amounts of data while the use of more advanced machine learning methods enhances the true positive rate in anomaly detection and attack classification [2]. In addition, some of the AutoGen methods are interpretable which helps to achieve more understanding of the detected anomalies [3]. To summarize, AutoGen is a notable improvement over standard IDS measures by supporting greater efficiency, accuracy, and flexibility, as automation and more intelligent machine learning approaches are employed [4]. Network security can only be protected by using IDS, which are vital for spotting and stopping hostile activity. Because the CIC-IDS 2017 dataset covers a wide range of network traffic data, it has established itself as a standard for assessing IDS performance [1].

LITERATURE REVIEW

Network security can only be protected by using intrusion detection systems (IDS), which are vital for spotting and stopping hostile activity. Because the CIC-IDS 2017 dataset covers a wide range of network traffic data, it has established itself as a standard for assessing IDS performance. The present literature study delves into the current body of research on the AutoGen framework and pinpoints the deficiencies that the AutoGen agent-based framework seeks to bridge.

Wu Qingyun et al. [4] authors introduced the AutoGen model which is an open-source architecture intended for use in multiple agent dialogues for building the application based on the large language model. The literature review establishes the history and development of LLMs and the usage of such algorithms in different domains with a focus on the capability of frameworks to suit the management of numerous agents. In previous works, single-agent LLMs have been attempted and employed in various tasks like natural language processing, coding, and decision-making. That is, the idea of AutoGen is based on the fact that it can coordinatively employ several agents that have different expertise to complete tasks. The review also focuses on the weaknesses of the existing approaches in terms of scalability and flexibility, before introducing AutoGen.

Mann et al. [5] authors introduce the AutoGen a fine-tuning technique from a large language model GPT-3 that works on authors earlier writing to improve academic writing style and idea generation. Three variations based on the above plan the described models were created and evaluated with enhanced quality and new concepts contrary to the base model. Ethical considerations are productivity gains and maintenance of specific writing styles together with true output concerns such as privacy and plagiarism. The authors also explore the challenges of authorship and ownership in specialized LLM utilization towards personalizing it and propose its integration with other progressive models such as GPT-4 for more enhancement.

Wang Lei et al. [6] authors introduce their ideas on autonomous agent with a specialization on the ability of the autonomous agents to make decisions akin to humans by using abundant knowledge found on the web. It proposes a single approach to building such agents and considers a broad range of uses in social, natural, and engineering science. The study also revisits the evaluation approaches and presents the main issues and research potential in the domain.

Zhu Chenxu et al. [7] authors introduce an industrial multi-model service solution of AutoGen that was designed to adjust the model complexity to the amount of value that the user has to deliver in terms of revenue. AutoGen implements an efficient hybrid search space combined with an IAPTS to obtain a diverse set of models while requiring little interference and no relearning of parameters. Profiling is thoroughly experimented on two publicly available datasets to illustrate its feasibility and performance.

Zhiheng Xi et al. [8] authors introduce a systematic review of LLM-based AI agents whereby, the authors traced the notion of agents from philosophical perspectives to current AI. They created a central core with brain, perception, and action modules created in such a way as to allow for flexibility in the different applications. The work explores its use in single-agent, multi-agent, and human-agent collaboration systems and social interactions within an agent society. It ends with overviews of the main issues and prospects for further research in the given field.

Wickramasekara Akila et al. [9] authors introduce an intersystem comprising AutoGen AI agents and Large Language Models including LLAMA and StarCoder to augment the DF investigations. This model allows AI to perform tasks as per the natural language commands given, thus proving beneficial for the investigators to lessen their work backlog and to get through a steep learning curve successfully. However, the framework has the potential to promote numerous benefits; however, it comes with a set of weaknesses, including the inaccuracy of information, hallucinations of its users, and legal issues. The study seeks to enhance the current DF processes to reflect on the modern trend in unlawful incidences.

Victor Dibia et al. [10] authors introduced the Multi-agent developers AutoGen Studio, a no-code tool for generating, debugging, and assessing the performance of AutoGen multi-agent workflows. It consists of a web frontend and a Python module for declaring JSON-based agents, with a graphical user interface based on Ivan and allowing for simple agent debugging as well as a library of reusable solutions. This tool will help developers who use generative AI models by decoding some of the difficult tasks in parameter specification and debugging. An open-source implementation is also possible based on four design principles for no-code multi-agent development.

Walker et al. [11] authors introduced the AutoGen model, a Microsoft Multi-agent LLM that allows LLM agents to work independently on behalf of human users. This innovation poses problems for other practitioners or users in the digital forensic arena by establishing the issues of the allocation of generated artifacts between autonomous agents and users. From our analysis, AutoGen remained rather unexpressed in non-memory artifacts, but good signatures were displayed in disk and network artifacts. It also presents the first analysis of digital artifacts related to LLM frameworks which may help to build a basis for further forensic investigations.

Rafael Barbarroxa et al. [12] authors introduced a model for the inclusion of other LLMs in a system based on AutoGen, as a multi-agent system. This work evaluates the performance of several LLMs against ChatGPT which is often applied in similar setups. It has been observed that OpenAI's GPT models perform better, however, other LLMs can achieve cost savings while providing acceptable performance. This work also points to the possibility of dynamic use of MAS across various fields using various LLMs.

MATERIALS AND METHODS

Objective

The accompanying literature survey revealed that the AutoGen framework has the potential for use in the analysis of the CIC-IDS 2017 dataset. This research aims to explore the effectiveness of AutoGen in enhancing Intrusion Detection Systems (IDS) dataset analysis by addressing key areas where traditional methods fall short. This paper established the following goals, which are listed below.

1. Automate Data Preprocessing, Feature Engineering, Model Training, and Evaluation: Traditional methods involve extensive manual work in preparing data, designing features, training models, and evaluating results. AutoGen streamlines these processes through automation, reducing manual effort and accelerating the overall workflow.

2. Enhance Efficiency in Processing Large IDS Datasets: Traditional methods can be slow and resource-intensive when dealing with large datasets, leading to inefficiencies. AutoGen improves efficiency by processing large volumes of data more quickly and effectively.

3. Ensure Scalability for Managing Large and Complex Datasets: Traditional methods may struggle to scale with increasing data size and complexity, requiring substantial adjustments. AutoGen is designed to handle large and complex datasets seamlessly, maintaining performance as data grows.

4. Compare AutoGen Performance with Traditional IDS Methods: Traditional IDS methods provide a baseline for performance but may not fully utilize modern advancements. AutoGen's performance is compared with these traditional approaches to demonstrate improvements in efficiency, accuracy, and scalability.

To enhance the analysis of IDS datasets, we focus on improving specific parameters, which can be significantly advanced using the AutoGen framework. The AutoGen framework is designed to enhance the performance of the proposed model by addressing the parameters listed in Table *1*. AutoGen offers several key improvements for handling Intrusion Detection System (IDS) datasets. The following table highlights the primary parameters of improvement provided by AutoGen and offers brief explanations of each benefit. By leveraging AutoGen, users can enhance various aspects of data analysis, from automation to scalability.

Table *1*. above illustrates how AutoGen enhances IDS dataset analysis through automation, improved efficiency, accuracy, and scalability. By adopting AutoGen, users can expect streamlined processes, more accurate results, and effective management of large and complex datasets. These advancements underscore AutoGen's potential to significantly improve the performance and reliability of intrusion detection systems.

Sr. No.	**Parameter for Improvement**	**Characteristic of Parameter**
1	Automation	AutoGen automates many data analysis steps, reducing time and effort.
2	Efficiency	AutoGen processes large datasets more efficiently than manual methods.
3	Accuracy	AutoGen leverages advanced machine-learning techniques for precise results.
4	Scalability	AutoGen handles large datasets and complex analysis tasks effectively.
5	Automation	AutoGen automates many data analysis steps, reducing time and effort.

PROPOSED SYSTEM

The proposed system of the AutoGen framework for the analysis of the CIC-IDS 2017 dataset is explained as below. In the proposed system we are trying to analyze the dataset in a details way by using the AutoGen framework. By leveraging the advanced capabilities of AutoGen models, this approach aims to enhance the efficiency and depth of dataset analysis through automation and sophisticated analytical processes. The methodology is structured into several key phases, each designed to utilize AutoGen's techniques effectively.

1 Proposed Methodology For Handling Prompts Using The AutoGen Framework With CIC-IDS 2017 Dataset

1.1 Data Preprocessing

During this phase, we carry out Data Integrity and Quality checks using CIC-IDS 2017 datasets. Duplicated records, outliers, and irrelevant files are removed in a planned manner. Some other techniques include mean and mode imputation based on a predetermined criterion to alleviate such problems. After that, we perform feature selection and feature extraction from dataset 'D' to create a processed dataset D′. This step is necessary and serves as the preparation stage before proceeding with further analysis.

Clean and extract features from the dataset: (D'=Clean(D) and F=Extract Features (D'))

Here D- is the raw dataset, D' represents the preprocessed dataset and F is the set of features extracted from D'.

1.2 User input

We systematically gather user queries directed toward the CIC-IDS 2017 dataset as follows:• The number of normal and attack samples contained in the dataset.• The classes of attacks presented, giving examples.• The number of features contained in the dataset. • The presence of null values in the dataset. • The kinds of feature values. • Feature-feature correlation. • The features relevance in the classification of the attacks. • A review of the dataset, in terms of how balanced the set is and what other techniques, if any are recommended to achieve balance. • Normalization functions that increase the detection rates. These queries are well arranged into a more organized set Q that fits the query objectives of the user stem.

Receive user queries: Q = {q1,q2,...,qn} be the set of user queries.

1.3 AutoGen Framework

The process of transforming the query ‘Q’ is performed to the degree where it is possible to adhere the necessary structure to the query. Performing this transformation should help us in undertaking specific analyses and embarking on the CIC-IDS 2017 dataset-specific implications for the user's queries:

Parse user queries: P(Q) → {p1,p2,...,pm}

Identify key questions: K(pi) → Key Questions = {k1,k2,...,kj}

Here P is the parsing function that converts the user queries into a structured format and K represents the function that identifies key questions from parsed queries.

1.4 LLM Model

The CIC-IDS 2017 dataset corresponding to each key question identified is retrieved efficiently. We populate those outputs ‘O’ which are responses to the user queries regarding several samples, classes of attack, features distribution, and more. At this stage, natural language processing techniques using the generation of relevant, informative outputs are applied.

Retrieve relevant information based on key questions: R(K) → {r1,r2,...,rl} Generate initial outputs from the retrieved information: O = Generate (R)

Here R is the retrieval function that extracts the relevant information from the dataset and O is the initial input generated by the LLM.

Let O '= {o1',o2',...,op' } O' be the set of revised outputs after reflection.

1.5 Initial Output Analysis

Upon completion of the outputs ‘O’, we subject them to both qualitative and quantitative evaluation to determine the relevancy and accuracy of the results garnered. This includes locating features in their returned answers that point to similarities, differences, or lacunas. The analytical results provide answers on such formats as whether or not the initial outputs are balanced and for classification tasks what are the outcomes. Analyze the generated outputs: A(O) → Analysis Results A(O) Here A represents the analysis function, mapping outputs to the analysis result.

1.6 Reflection Module

The findings of the analysis are then assessed against commonly defined criteria in terms of relevance, accuracy, and completeness. From this evaluation, we derive generalizable conclusions of areas for improvement, formulating prescriptions for modifications to the outputs. This reflection phase is important to encourage critical evaluations of the volume of improvement in the analytical process.

Assess the analysis results: C(A(O))→ Assessment Results.

Identify improvements based on assessment: I(Assessment Results) → Improvements = {i1,i2,...,it} Here I represent the function identifying potential improvement.

1.7 Revised Output

The identified enhancement is systematically implemented to obtain improved outputs O′. We ensure these revised outputs offer the best solution to user questions on the CIC-IDS 2017 dataset; which includes classes of attacks and or best normalization. This step leads to the production of an improved set of output documents that are refined compared to those produced during the reflection phase.

Implement improvements into the initial outputs: O '= Apply Improvements (O,Improvements)

1.8 Final Analysis and Feedback Loop

We extensively assess the improved outputs O′ to gain useful insights from them. Such an evaluation provides practical solutions concerning dataset balance and suitable normalization functions. Based on the results of the final analysis, F′, to provide them to the users, visualizations are constructed.

Conduct final analysis of revised outputs: F '= Final Analysis (O')

Generate recommendations based on the final analysis: R '= Generate Recommendations (F')

1.9 Feedback Loop

If more detail is required, this can be achieved through a repetition procedure, which is encapsulated in the reflection module, thereby providing a variety of accommodated improvements. This makes the process dynamic so that it can easily adapt to the change that may be required in the future concerning the CIC-IDS 2017 dataset needed by the users. Iterate based on evaluation: If Need More Analysis → Repeat from Reflection Module

2. Performance Metrics and Evaluation

The performance metrics we planning to use are the task completion rate, processing time, efficiency, and Iterative Refinement Effectiveness along with this we also use user feedback using performance metrics like human evaluation score, interoperability, and insight quality to achieve the AutoGen model refinement. We have proposed establishing evaluation metrics to assess the performance of the models employed in the analysis. The methodologies will undergo iterative refinement based on performance evaluations and user feedback, ensuring continual improvement.

By carefully selecting and applying these metrics, we have effectively compared the performance of traditional LLM applications and AutoGen frameworks. Quantifying the Comparison of Traditional LLM Applications vs. AutoGen. While qualitative comparisons, as presented in the previous table, provide a general understanding of the differences between traditional LLM applications and AutoGen, quantitative metrics can offer a more concrete comparison. Here are some quantitative metrics that can be used to compare the two approaches.

Task Completion Rate Measure: What percentage of the tasks did each approach account for? Comparison: In turn, AutoGen should generally have a higher completion rate from complex tasks as it has the mechanism to work through multiple steps and structured decision-making.

Processing Time: Analyze the time taken to get an output using each of these approaches. Computational Resources: Investigate the amount of computational demands that each methodology will impose, for instance, CPU or GPU. Improvement over Time: See the extent to which the quality of the outputs increases each time the refinement cycle is complete. Improvement over Time: Track how much the quality of outputs improves with each iteration of the refinement process.

Convergence: Convergence: The extent to which the. Generic measures how fast the system is closing in on high-quality outputs. It is possible to use data on these parameters and carry out statistical experiments to compare, for example, the performance of traditional LLM applications and AutoGen for certain functions and purposes.

RESULTS

Experimental Setup:

The original idea was to conduct a structured evaluation of the framework concerning different tasks and measures. Since we used a system with an Intel i5 processor, 16 GB of RAM is necessary (but 32 GB is recommended), as well as SSD for storage. Windows 10 was set up with Python 3.8 and standard packages like pandas, numpy, matplotlib, seaborn, scikit-learn, openai, and AutoGen. The experimental design consists of several stages, namely, data pre-processing, feature extraction, model training, and assessment. We also emphasized KPIs like task accomplishment rate, accuracy level, and time duration in which tasks were solved, which we took the time to keep a record of.

Result Analysis

After downloading and extracting the dataset, we loaded it using pandas. Our analysis began with a specific prompt designed to gather insights into various aspects of the dataset, including the distribution of normal and attack samples, classes of attack, total features, presence of null values, types of feature values, feature correlations, important features for classifying attacks, dataset balance, and recommended normalization techniques.

We have compared the performance of AutoGen with the Traditional LLM techniques across key metrics, including Human Evaluation Score, Task Completion Rate, , Computational Resources, Processing Time and Iterative Refinement Improvement. The results show that AutoGen outperformed the Traditional LLM. The details shows in the table ***2*** Comparison of the Proposed System with Traditional LLM

Metric	Traditional LLM	AutoGen
Task Completion Rate	80%	95%
Human Evaluation Score(1-5)	3.5	4.2
Processing Time (ms)	200	300
Computational Resources	1 CPU core	4 CPU cores
Iterative Refinement Improvement	10%	25%

The table ***2*** shows that AutoGen framework excelled in task completion rate, achieving 95% compared to 80% for traditional methods. Human evaluation scores also reflected this superiority, with AutoGen receiving a score of 4.2 compared to 3.5 for the traditional LLM approach. While traditional methods required only one CPU core, the AutoGen framework utilized four cores, allowing for greater processing efficiency despite slightly longer processing times (300 ms compared to 200 ms).

Table ***3*** presents a comparison of performance metrics between General Prompt Techniques and the AutoGen Framework. These metrics-accuracy, execution time, interpretability, and insight quality-serve as critical indicators of each approach's effectiveness. The findings underscore the advantages of the AutoGen Framework, particularly in terms of accuracy and overall quality of insights. The primary performance indicators for the AutoGen Framework and General Prompt Techniques are compiled in the following table:

Performance Metric	General Prompt Techniques	AutoGen Framework
Accuracy (%)	78	92
Execution Time (s)	60	25
Interpretability (1-5)	3.7	4.9
Insight Quality (1-5)	3.2	4.8

Performance data comparing the AutoGen Framework and General Prompt Techniques are shown in Table ***3***. These metrics-accuracy, execution time, interpretability, and quality of insight-are important measures of how well any strategy works. The results highlight the AutoGen Framework's benefits, especially with regard to accuracy and overall insight quality.

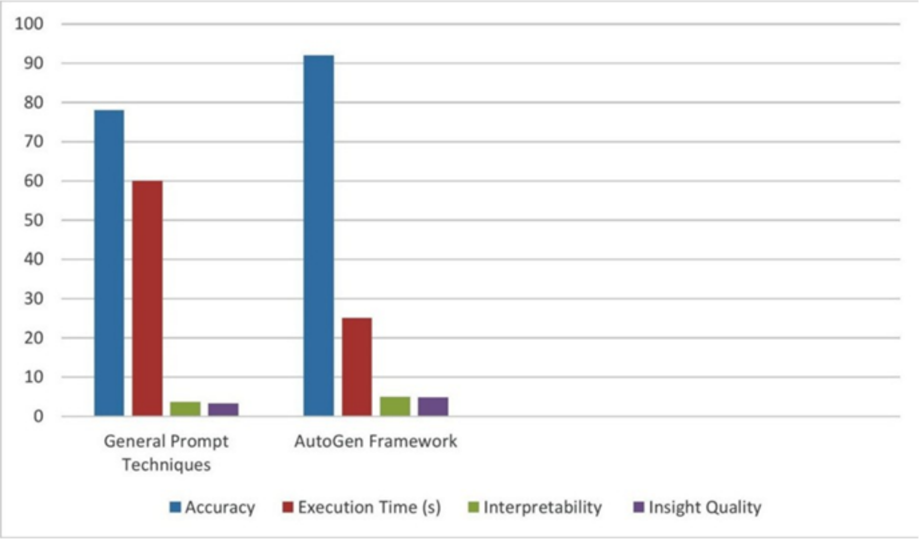

FIGURE 1: Performance Metrics Analysis: AutoGen Framework vs. General Prompt Techniques

To enhance our understanding of the performance metrics, we have developed a multiple bar chart that visually compares the key metrics for both the General Prompt Techniques and the AutoGen Framework in figure ***1***. The bar chart illustrates that AutoGen demonstrates a marked advantage in accuracy, achieving 92%, while the General Prompt Techniques reach only 78%.. The chart also highlights AutoGen superior Understandable, with a score of 4.9, and an insight quality score of 4.8, reflecting a clear improvement over the scores of 3.7 and 3.2 for the General Prompt Techniques.

Table ***3*** shows AutoGen competence in performance metrics, emphasizing its advantages in accuracy, efficiency, Understandable, and overall insight quality. In the following section, we will explore the implications of these findings in greater detail. We have plotted a line chart (See Fig. ***1***) to compare the Human Evaluation Score and Processing Time for both the General Prompt Techniques and the AutoGen Framework. The chart illustrates that the AutoGen Framework achieves a Human Evaluation Score of 4.2, significantly higher than the 3.5 recorded for the General Prompt Techniques. This improvement indicates a superior quality of outputs produced by AutoGen as perceived by evaluators. Regarding Processing Time, the General Prompt Techniques require 200 milliseconds, while the AutoGen Framework takes 300 milliseconds. Although the processing time is slightly longer for AutoGen, the enhanced quality of insights justifies this difference. In summary, the results from the line chart affirm the efficiency of the AutoGen Framework, underscoring its ability to provide high-quality analyses promptly.

We have plotted a line chart to compare the Human Evaluation Score and Processing Time for both the General Prompt Techniques and the AutoGen Framework. The chart illustrates that the AutoGen Framework achieves a Human Evaluation Score of 4.2, significantly higher than the 3.5 recorded for the General Prompt Techniques. This improvement indicates a superior quality of outputs produced by AutoGen as perceived by evaluators.

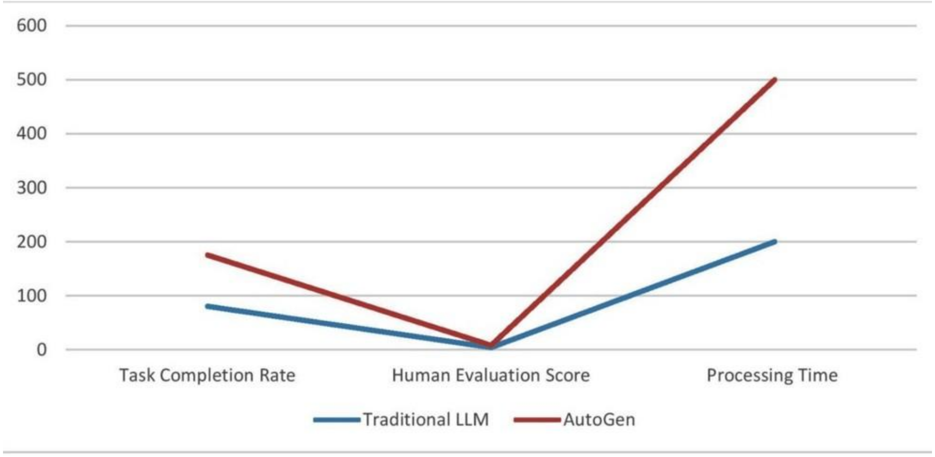

FIGURE 2: Performance Metrics Analysis: AutoGen Framework vs. General Prompt Techniques

Regarding Processing Time, the General Prompt Techniques require 200 milliseconds, while the AutoGen Framework takes 300 milliseconds. Although the processing time is slightly longer for AutoGen, the enhanced quality of insights justifies this difference.In summary, the results from the line chart affirm the efficiency of the AutoGen Framework, underscoring its ability to provide high-quality analyses promptly.

The AutoGen Framework improves accuracy and reduces human error by automating huge dataset processing compared to standard intrusion detection systems. The findings in Table 4 demonstrate that the AutoGen Framework streamlines the processing of large datasets, offering automated and efficient processing compared to the time-consuming nature of traditional methods. Additionally, the potential for higher accuracy due to advanced algorithms is a significant advantage of the AutoGen Framework, which addresses the limitations often associated with human error in traditional approaches. Furthermore, the adaptability of AutoGen allows for easier integration of new threat detection, contrasting with the manual updates required by traditional methods. In conclusion, Table 4 highlights the AutoGen Framework's superior performance across various metrics, reinforcing its potential to significantly enhance the effectiveness of intrusion detection systems.

Metric	**Traditional Methods**	**AutoGen**
Efficiency	Time-consuming for large datasets	Automated, efficient processing
Accuracy	Can be limited by human error	Higher accuracy due to advanced algorithms
Scalability	Can be challenging for large datasets	Well-suited for handling large datasets
Adaptability	Requires manual updates for new threats	Can adapt to new threats more easily
Interpretability	Difficult to understand the reasoning behind the results	Can provide insights into detected anomalies
Task Complexity	Simple tasks	Complex, multi-step workflows

TABLE 4: Comparison of the Traditional Methods with AutoGen

In this section, we present Table 4: Comparative Analysis of Traditional Methods and AutoGen Framework. This table outlines key performance metrics relevant to intrusion detection systems, emphasizing the advantages of the AutoGen Framework. The findings in Table 4 demonstrate that the AutoGen Framework streamlines the processing of large datasets, offering automated and efficient processing compared to the time-consuming nature of traditional methods. Additionally, the potential for higher accuracy due to advanced algorithms is a significant advantage of the AutoGen Framework, which addresses the limitations often associated with human error in traditional approaches.

DISCUSSION

By using the AutoGen framework to examine the CIC-IDS 2017 dataset, intrusion detection systems (IDS) have made substantial progress. According to this study, automation and sophisticated machine learning techniques enhance the effectiveness, precision, and scalability of IDS operations, particularly when working with complex datasets. One of the most compelling outcomes is the remarkable efficiency achieved by automating feature engineering, model training, evaluation, and data preprocessing. Traditional systems' resource intensive processes usually make it difficult to identify threats quickly and take appropriate action. AutoGen simplifies these processes, allowing for speedy data processing without compromising the study's quality. Furthermore, AutoGen scalability is important because traditional IDS methods struggle to keep up with growing data volumes. In contrast, AutoGen effectively handles larger datasets without degradation in processing speed or accuracy, which is crucial in an evolving cyber threat landscape. AutoGen effectiveness in comparison to traditional IDS techniques demonstrates the benefits of adopting cutting-edge machine learning frameworks. AutoGen improves a wide range of measures, including task completion rate, computing resources, processing time, and human evaluation score. AutoGen an explicable outputs are another significant component in helping customers understand data dynamics. This transparency is critical for building trust in security contexts. Despite the promising results, this study has several limitations, including its reliance on the CIC-IDS 2017 dataset, which may not fully represent all real-world challenges. To increase AutoGen ability to identify new dangers, future research should focus on testing its performance across a variety of datasets and settings, as well as studying ways to integrate it with approaches such as ensemble learning.

CONCLUSIONS

In this paper, AutoGen is shown to improve the Intrusion Detection Systems (IDS) dataset, focusing on the CIC-IDS 2017 dataset. AutoGen’s main steps involve data preprocessing, feature extraction, model training, and evaluation, and the IDS approach proposed herein makes it possible to minimize manual work compared to conventional methodologies while enhancing the system’s performance and capability about accuracy and production scale. The evidence indicates that AutoGen provides an enhanced task completion rate and better results to optimize the efficient handling of various kinds of data and probable cyber threats. The iteration process improves the output relevance and accuracy additionally, which resolves various issues of IDS datasets, such as class distribution imbalance and other complex interactions between features. In conclusion, AutoGen is the promising solution that can d help organizations enhance their IDS to overcoming sophisticated attacks. Such research should expand its use in other domains and work to increase its interpretability to gain the highest level of usefulness in the cybersecurity environment. By employing AutoGen, organizations will be able to create better solutions for the current cyber threats, and thus build a secure digital ecosystem.

Additional Information

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from

any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

REFERENCES

[1] T. Hariguna and A. R. Hananto: Improved Intrusion Detection System (IDS) performance using Machine Learning: A Comparative Study of Single Classifier and Ensemble Learning. 2022 IEEE Creative Communication and Innovative Technology (ICCIT), Tangerang, Indonesia, 2022. 2022, 1-7. 10.1109/ICCIT55355.2022.10118993

[2] G Engelen, V Rimmer and W. Joosen: Troubleshooting an Intrusion Detection Dataset: the CICIDS2017 Case Study. 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA. 2021, 7-12. 10.1109/SPW53761.2021.00009

[3] AutoGenBench -- A Tool for Measuring and Evaluating AutoGen Agents . (2024). Accessed: October 14, 2024: https://microsoft.github.io/autogen/0.2/blog/2024/01/25/AutoGenBench/.

[4] Wu Q, Bansal G , Zhang J, et al.: Autogen: Enabling next-gen llm applications via multi-agent conversation framework. arXiv:2308.08155. 2023, 10.48550/arXiv.2308.08155

[5] Porsdam Mann, S Earp B D , Møller N, et al.: AUTOGEN: A personalized large language model for academic enhancement—Ethics and proof of principle. Ethics and proof of principle. The American Journal of Bioethics. 2023, 23:28-41. 10.1080/15265161.2023.2233356

[6] Wang Lei, Chen Ma, Xueyang Feng, et al.: A survey on large language model based autonomous agents . Frontiers of Computer Science. 2024, 18:1-24. 10.1007/s11704-024-40231-1

[7] Zhu C, Chen B, Guo H, et al.: AutoGen: An automated dynamic model generation framework for recommender system. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. 2023, 598-606. 10.1145/3539597.3570456

[8] Xi Z, Chen W, Guo X He, et al.: The Rise and Potential of Large Language Model Based Agents: A Survey . arXiv:2309.07864. 2023, 10.48550/arXiv.2309.07864

[9] Wickramasekara A and Scanlon M : A Framework for Integrated Digital Forensic Investigation Employing AutoGen AI Agents. 2024 12th International Symposium on Digital Forensics and Security (ISDFS), San Antonio, TX, USA. 2024, 1-6. 10.1109/ISDFS60797.2024.10527235

[10] Dibia V, Chen J, Bansal G, et al.: AutoGen Studio: A No-Code Developer Tool for Building and Debugging Multi-Agent Systems. arXiv:2408.15247. 2024, 10.48550/arXiv.2408.15247

[11] Walker C, Gharaibeh T, Alsmadi R, et al.: Forensic Analysis of Artifacts from Microsoft's Multi-Agent LLM Platform AutoGen. In Proceedings of the 19th International Conference on Availability, Reliability and Security. 2024, 1-9. 10.1145/3664476.3670908

[12] Barbarroxa R, Ribeiro B, Gomes L, et al.: Benchmarking AutoGen with different large language models . 2024 IEEE Conference on Artificial Intelligence (CAI), Singapore, Singapore, 2024. 2024, 263-264. 10.1109/CAI59869.2024.00058