A novel approach to parallel anomaly detection: application in cybersecurity

Ziani Zineb, Emad Nahid, Bouaziz Ahmed

A Novel Approach to Parallel Anomaly Detection: Application in Cybersecurity

1st Ziani Zineb
University of Paris Saclay
Numeryx / Li-PaRAD / MDLS
Paris, France
zineb.ziani@universite-paris-saclay.fr

2nd Emad Nahid
University of Paris Saclay
Li-PaRAD / MDLS
Paris, France
nahid.emad@uvsq.fr

3rd Bouaziz Ahmed
Numeryx Company
Paris, France
aa.bouaziz@numeryx.fr

***Abstract*—Introducing the Scalable Anomaly Detection with UC2B framework, this paper addresses the critical task of identifying unusual patterns in data, crucial for effective cyber threat defense. By leveraging ensemble learning methods and the parallel processing capabilities of the Unite and Conquer approach, the framework demonstrates its proficiency in handling large datasets. It strives to offer computational efficiency, scalability, and high accuracy in real-world applications. Notably, this paper places special emphasis on the diversity of components and acknowledges their substantial influence on the overall framework functionality. It encompasses features such as fault tolerance, adaptability to various architectures, and efficient load balancing. Experimental validation on the Ruche Cluster within the realm of cybersecurity provides valuable insights into its potential in detecting anomalies.**

***Index Terms*—Anomaly Detection, Linear Algebra, Unite and Conquer Approach, Machine Learning, High performance computing, Ensemble learning, UC2B, Cybersecurity.**

I. Introduction

Anomaly detection, a critical aspect of data analysis, has gained significant prominence in various domains due to its potential to identify patterns or behaviors that deviate significantly from normal or expected observations. It finds applications in a variety of domains, helping to identify unusual behaviors or events that may indicate potential issues or opportunities [8].

For instance, in the financial domain, anomaly detection plays a crucial role in detecting fraud and suspicious financial activities [5]. By identifying unusual transactions, atypical spending patterns, or fraudulent behaviors, anomaly detection enhances the security and protection of financial assets. Similarly, in the manufacturing industry, anomaly detection is leveraged to monitor production processes and identify failures or unexpected variations, enhancing product quality, optimizing operations, and minimizing downtime [22]. Moreover, in the healthcare domain, anomaly detection aids in the detection of unusual symptoms in patients, the identification of rare or emerging diseases, and the analysis of medical images for accurate diagnosis and timely interventions [25].

In the realm of cybersecurity, anomaly detection serves as a vital tool for safeguarding computer systems against malicious attacks. By detecting abnormal behaviors on networks, data breaches, hacking activities, and intrusion attempts, anomaly detection plays a crucial role in threat detection and prevention [23]. It enables security professionals to identify and respond to real-time attacks, enhancing system security and minimizing the impact of cyber threats.

Advancements in anomaly detection, driven by the availability of large-scale datasets and sophisticated machine learning algorithms like unsupervised learning, deep learning, and ensemble methods, have brought about significant improvements. These improvements pertain to anomaly detection performance in the realm of cybersecurity. However, the increasing complexity and size of modern datasets necessitate substantial computation power to efficiently process and analyze the data [13]. Deep learning models, for example, require high-performance GPUs or specialized hardware accelerators for training and fine-tuning. On the other hand, real-time anomaly detection necessitates rapid analysis of incoming data streams, relying on the processing speed and scalability of modern hardware. Therefore, to fully harness the potential of these advanced techniques and effectively detect anomalies in complex data, organizations must invest in powerful computational resources capable of handling the intensive processing demands of modern anomaly detection methods.

To tackle these challenges head-on, we have developed the UC2B framework, in a previous work, an acronym for Unite and Conquer with Bagging and Boosting. This approach employs a collaborative training approach for co-methods that mutually enhance their performance through sequential iterations. This process involves co-methods boosting one another within the same iteration. Each co-method undergoes training on multiple parallel bags. After this iteration, a global boost occurs by reintroducing misclassified instances (FP/FN) into the bags, leading to improved detection in the subsequent iteration. This dual boosting strategy yielded an impressive detection rate of up to 99% in the field of cybersecurity, notably on the UNSW-NB15 benchmark dataset. However, these experiments have highlighted scalability concerns. The framework faced challenges in handling the computational demands of large-scale datasets, resulting in inefficiencies related to memory usage and computational complexity, particularly as parallelism was confined to within the co-methods. Additionally, the duration of model training exceeded one week, making its practical application in real-world scenarios

nearly unfeasible.

In terms of contributions, this work leverages the insights of the UC2B framework, introducing the 'Scalable Anomaly Detection with UC2B' framework, referred to as 'Parallel UC2B'. This has been validated on the Ruche Cluster and has demonstrated promising results. This paves the way for its seamless integration into an existing cybersecurity threat detection and remediation platform. This effort specifically emphasizes cybersecurity and capitalizes on the UNSW-NB15 dataset [20]. The integration of the parallel UC2B framework aims to advance methods and address abnormal behaviors. Benchmark evaluations are pivotal for performance assessment.

These contributions collectively strengthen the field of anomaly detection and extend its applicability across various domains. They encompass:

- Introduction of a refined configuration featuring multi-level parallelism, a fusion of bagging and boosting, augmented by a restarting strategy inspired by the unite and conquer method for anomaly detection.
- Integration a rich diversity of components, encompassing a wide range of ML models. This diversity extends to features such as inherent load balancing, distributed computation capabilities, and a fault-tolerant implementation strategy.
- The augmentation of parallel co-methods, alongside the inclusion of parallel bags of variable sizes, has significantly bolstered the framework's capacity. This empowers the system to proficiently process extensive databases while maintaining high efficacy across a broad spectrum of anomaly detection applications.
- Implementation of a parallel framework designed to harness the performance capabilities of high-performance computing architectures, including execution time and scalability.
- Rigorous validation of the framework's efficacy through a series of experiments executed on Ruche Cluster.
- Specialized focus on the application of the framework within the cybersecurity domain, leveraging the widely recognized UNSW-NB15 dataset for evaluation.

II. Software Architectures of Parallel UC2B

In many scientific disciplines, data generation now surpasses computational capacities. The effective processing of this massive volume of data relies on the integration of modeling, analysis, and high-performance computing techniques [24]. This interplay presents various scientific and technological challenges. However, these challenges, prevalent across diverse application domains, share a common foundation in applied mathematics (including linear algebra and statistics) and artificial intelligence, encompassing machine learning methods, as well as high-performance computing techniques.

In the realm of cybersecurity, security breaches are evolving to become more subtle and sophisticated, resulting in extended investigation times for alerts and a heightened need for discernment between authentic and false alarms. Expertise and the time invested therein are paramount within a Security Operations Center (SOC). For example, swiftly discerning the validity of 'false alerts' can prove pivotal in a security framework [15].

This undertaking seeks to contribute to the resolution of these challenges, exemplified through practical applications of data analysis in securing information systems within organizations, such as advanced technology enterprises.

As outlined in the state-of-the-art section, the application of the Unite and Conquer approach to Ensemble Learning methods, called UCEL, is another anomaly detection technique proposed by Diop et al. in [9], [10]. In this paper, we propose an extension of UCEL which improves its performance. To distinguish this extension from UCEL, we call it UC2B for Unite and Conquer with Bagging-Boosting. The presence of several levels of boosting as well as that of multi-level intrinsic parallelism in UC2B partly explain its better performance relative to UCEL. Other characteristics such as the heterogeneity of its components, its fault tolerance as well as its potential for load balancing make UC2B a technique very well suited to recent parallel and/or distributed architectures.

A. Unite and Conquer Approach

"Unite and Conquer" embodies a problem-solving paradigm that intricately orchestrates the collaboration of multiple iterative methods, or co-methods, to collectively address a given problem. Primarily applied within the realm of linear algebra, it proves exceptionally valuable in resolving expansive, sparsely populated linear systems, along with eigenvalue predicaments [12]. By aggregating intermediate outcomes from each co-method, this strategy expedites the convergence of the overall method, hastening the realization of a solution. Central to its efficacy is the strategic restarting approach, playing a pivotal role in providing a better starting point for each new cycle of the co-methods. This iterative refinement ensures that subsequent cycles are better poised for success, ultimately enhancing the overall convergence of the approach. Co-methods exchange their intermediate calculated solutions to discern the most effective restarting condition for each cycle, culminating in swifter global convergence.

This approach holds several advantages. Its intrinsic attributes, including multi-level parallelism, robust fault tolerance, adaptability to component heterogeneity, asynchronous communication capabilities, and inherent load balancing potential, render it eminently suitable for deployment in cutting-edge computational architectures. The "Unite and Conquer" approach optimally leverages these features, enabling the judicious allocation of computational resources, leading to heightened efficiency and the realization of parallel processing benefits. This not only accelerates the resolution of complex problems but also maximizes the utilization of available resources.

The Unite and Conquer algorithm can be expressed in a mathematical form as the following. Let P be the large linear algebra problem to be solved, $L_1, L_2, ..., L_l$ be a set of iterative methods that can solve P, I_i^k the the initial condition (with

$k = 0$) and restarting condition (with $k > 0$) of L_i, and θ be the threshold value. Let f be a function defining the restarting strategy according to the intermediary results $(S_1^k, ..., S_\ell^k)$ with S_i^k the approximated solution obtained by L_i at the end of i-th iteration/cycle. An algorithm of Unite and Conquer can be defined as follows:

Algorithm 1 Unite and Conquer Algorithm

Initialize Choose a starting matrix $[I_1^0, \ldots, I_\ell^0]$, let $k = 0$.
For $i = 1$ to ℓ **do in parallel**
Compute S_i^k by applying L_i to P with initial conditionI_i^k.
If S_i^k is sufficiently accurate, STOP all ℓ process and return S_i^k as the solution of P.
Share S_i^k information with all other processes j $(j = 1, \ldots, \ell$ and $j \neq i)$.
Update and Restart $[I_1^{k+1}, \ldots, I_l^{k+1}]$=$f(S_1^k, \ldots, S_l^k)$ and increment k.

At its essence, this approach is characterized by its simple yet versatile conceptual framework. It can be applied to a range of iterative methods, as illustrated in this paper through a practical example. Here, we explore the integration of various boosting techniques within the specific context of bagging methodologies, showcasing a second level of parallelism. This simultaneous approach enhances the model's adaptability and performance across diverse datasets and scenarios.

B. Parallel UC2B Insights

The proposed parallel UC2B framework is designed to enhance the accuracy and efficiency of anomaly detection, specifically addressing the challenges within the 'UC2B' framework. This is achieved by integrating the Unite and Conquer problem-solving approach with Bagging and Boosting techniques, and introducing multi-level parallelism. The aim is to improve the accuracy and speed of anomaly identification. The proposed approach operates through an iterative process, promoting collaboration among parallel co-methods (Machine learning models). Over multiple training cycles, the performance of these co-methods is refined, leading to a convergence state where substantial and stable performance improvements are observed. Notably, each co-method undergoes parallel training alongside its counterparts, with exposure to multiple parallel bags of the training set, highlighting a multi-level parallelism approach.

Subsequently, the performance of these co-methods is rigorously evaluated using a validation set. Individually, the base methods play a critical role by sharing their misclassified data (False Positives and False Negatives) with other co-methods. Following the boosting principle, when a co-method misclassifies a sample from the validation set, adjustments are made to the weight assigned to that sample based on the co-method's performance metric at that specific iteration. This process increases the likelihood of selecting the sample to construct the subsequent cycles' training data, thus driving the iterative process forward. As a result, bags of the original training data size are generated from this boosted training dataset and subsequently utilized in the next cycle. This intricate process enables each co-method to gain valuable insights from their peers about critical data samples that pose challenges for accurate classification. This collective effort ultimately leads to the refinement and advancement of the models.

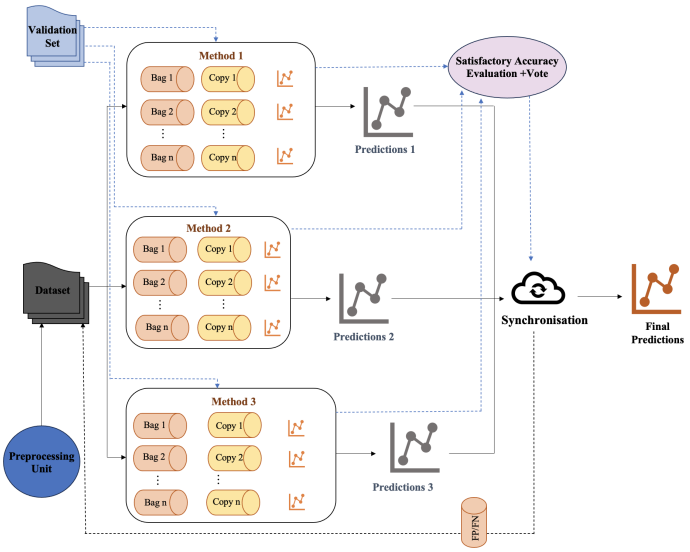

Fig. 1. Parallel "UC2B" Architecture.

Parallel UC2B exhibits improvements over both UC2B and UCEL. UC2B achieved a detection accuracy of 99%, but its very long training time makes its real-world application almost impossible. On the other hand, UCEL achieved an accuracy ranging from 93% to 98% depending on the databases containing different types of internal attacks, along with its lengthy training time.

The goal of parallel UC2B is to be versatile, addressing a wide spectrum of attacks, whether internal or external, as well as anomalies, all while maintaining a reasonable execution time for practical deployment. In tackling the challenge of detecting sophisticated threats and anomalies, we seek to leverage insights from each co-method on a dataset statistically identical in each iteration. Given that the Unite and Conquer approach learns the underlying global structure of data, this is why we provide the entire dataset to all methods in parallel. Each co-method then creates duplicates of itself and segments the dataset into multiple bags, training each copy of the co-method on a bag, see figure 1. This approach ensures that each co-method learns from the entirety of the dataset and collaborates synchronously with the other co-methods by sharing their outputs. In contrast, in UCEL, the dataset is divided into bags, with each bag exclusively assigned to a single co-method, limiting the number of bags. In parallel UC2B, the number of bags is independent of the co-methods, affording us the flexibility to have "n" bags for each method.

Unite and Conquer, as an approach, accommodates both synchronized and asynchronous communications. In this study,

our emphasis is placed on synchronous communications among co-methods as well as within the bags housed within these co-methods.

The collaborative mechanism among co-methods is grounded in bootstrapping, a method that entails diversified data treatment through the integration of bagging and boosting techniques. This integrated approach effectively balances the mitigation of bias and the management of variance. Specifically, bagging helps alleviate overfitting, while boosting counteracts underfitting. The training process integrates feedback from all co-methods, encompassing performance metrics and instances of false positives/negatives.

Furthermore, the inherent parallelism in the design allows for the utilization of computational resources. The independence between the number of bags and co-methods, unlike in UCEL, not only enhances computational efficiency but also enables scalability in handling larger datasets. It's noteworthy that the number of co-methods employed has a negligible impact on training duration, given the concurrent operation of the co-methods and efficient resource utilization. This aspect stands out as a distinct advantage in comparison to UC2B approach.

Within 'parallel UC2B', thread parallelism employs a dual-layered strategy, leveraging simultaneous thread execution for concurrent task handling. SIMD operations enhance computational speed by processing data batches in parallel. Efficient data transfer optimizes performance by utilizing duplicates to work on specific data subsets, reducing the need for frequent memory access. Collaborative data sharing among co-methods is crucial for handling misclassified data, with thread parallelism and SIMD operations driving this iterative process. Finally, efficient data Input/Output (I/O) ensures timely information exchange, supporting the boosting mechanism and leading to significant performance gains.

C. Algorithm and Implementation of Parallel UC2B

In the realm of machine learning, data analysis methods vary based on the available information, whether it's labeled, unlabeled, or imbalanced. In corporate environments, the majority of activities are routine, resulting in datasets skewed towards normal behavior. This abundance of normal data poses challenges for anomaly detection.

Traditional supervised and unsupervised methods may struggle with limited abnormal examples. Unsupervised techniques, while adept at handling imbalanced data, often focus solely on identifying deviations without delving into the underlying causes. Conversely, supervised methods excel when provided with balanced and labeled datasets, but this isn't always feasible in real-world scenarios.

To address these complexities, our anomaly detection framework strategically combines eight distinct methods: supervised (Logistic Regression and Multi-Layer Perceptron) LR and MLP, unsupervised (k-Nearest Neighbor and Elliptic Envelope) KNN and EE, and semi-supervised (Quadratic Discriminant Analysis, Linear Discriminant Analysis, Light GBM, and Extra Trees Classifier) QDA, LDA, LGBM and ETC. This comprehensive approach leverages the unique strengths of each method to create a robust anomaly detection system. LR and MLP make informed decisions based on labeled data, while k-NN and EE excel at identifying outliers without the need for labels. The QDA, LDA, LGBM, and ETC models add interpretability and effectively utilize both labeled and unlabeled data, showcasing their adaptability to semi-supervised learning scenarios. By integrating these diverse co-methods, our framework adeptly navigates the intricacies of unbalanced and unlabeled datasets, ultimately enhancing accuracy and reliability in real-world applications.

The ensemble of machine learning models, including supervised, unsupervised, and semi-supervised approaches, enables adaptability to diverse data scenarios, making it highly effective in detecting anomalies. Moreover, it addresses the challenges posed by unbalanced and unlabeled datasets, ensuring accurate and reliable results.

Our proposed approach initiates with a meticulous preprocessing of the dataset. This involves a series of crucial steps, including data cleaning, feature selection, as well as scaling and normalization procedures. Furthermore, from this refined dataset, distinct sets for training, validation, and testing are meticulously curated. The core tenet of our methodology involves integrating the principles of the "Unite and Conquer" paradigm into machine learning methods. This entails establishing meaningful correspondences between this paradigm and the intricate landscape of machine learning (co-methods). Here, the system matrix aligns with the original training dataset, while the subspace is meticulously constructed through Bootstrap sampling. The co-methods are LR, MLP, KNN, EE, QDA, LDA, LGBM and ETC each contributing its unique strengths to our anomaly detection framework. The inaugural phase involves simultaneous training of co-methods across multiple parallel bags. These co-methods are subsequently assessed using a validation set, employing accuracy as the metric. The outcomes of each co-method are amalgamated to forge a robust weighted voting classifier. This composite classifier is meticulously benchmarked against the individual co-methods performance and the predefined detection threshold. In contrast to the fixed iterations seen in Boosting ensemble methods, UC2B's iterations persist until the desired precision is achieved. It employs a combination process, replicated multiple times and connected sequentially to form a serial Boosted scheme. In cases where the established detection threshold is not met, a revival boosting step is implemented, focusing on the previous training data. Instances of false positives/negatives (FP/FN) are systematically gathered and then reintegrated into the dataset with enhanced weights. The training data for the next iteration of a co-method is carefully crafted by combining its most accurate training bag from previous iterations with the most widely recognized FP/FN instances. This iterative process continues until the predetermined threshold or specified number of iterations is satisfactorily achieved. Upon reaching the desired level of accuracy, the process concludes smoothly.

The algorithm can be defined as follow:

Algorithm 2 Parallel UC2B

Input:
Data set D.
Number of bags I.
Number of all process iterations n.
Number of learners M.
for $i \leftarrow 1$ to n **do:**
  for $j \leftarrow 1$ to M **do in parallel:**
    for $k \leftarrow 1$ to I **do in parallel:**
      $B_k \leftarrow$ Bags Bootstrap sample from D with replacement.
      $y_k \leftarrow$ Vector label issued L_j training on the bags B_k.
      Predictions[j] $\leftarrow$ Prediction using y_k.
  Sync and Share the results with all other processes.
  Check for desired accuracy; if met, stop all process n.
Restart by Updating the input data with adjusted sample weights and proceed to the next iteration.
Output:
Obtain the boosted predictions after the desired iterations.

In parallel UC2B Implementation, the framework embraces multi-level parallelism within the co-methods, encompassing both data parallelism and model parallelism. Each co-method undergoes concurrent training and generates copies corresponding to the chosen number of bags. Communication between co-methods and bags can be synchronous or asynchronous, effectively combining coarse-grain inter co-methods and fine-grain intra co-method parallelism. This approach also allows for the use of different co-methods, leveraging specialized hardware processors for optimized performance. While the article primarily emphasizes synchronous implementation, it acknowledges potential time loss due to synchronization. The algorithm's fault-tolerant nature ensures uninterrupted functionality even if one co-method is absent. Within this setup, co-methods (SNs) operate as computing servers, concurrently processing the dataset, further dividing it into bags and creating corresponding copies, introducing multi-level parallelism. The controller (CN) manages co-method and bags synchronization, evaluates their results, and selects the best outcomes from each bag for subsequent algorithmic cycles. This configuration, with its multi-level parallelism and coordinated decision-making, significantly bolsters the efficiency and accuracy of anomaly detection.

D. Integrating Parallel UC2B with SOAR Capabilities

A Security Orchestration, Automation, and Response (SOAR) platform is a comprehensive cybersecurity solution that integrates and streamlines the management of security alerts and incidents within an organization [26]. It combines four crucial elements: detection, orchestration, automation, and response (see Figure 2). Detection involves the monitoring of various data sources, including logs, network traffic, and security alerts, to identify potential security incidents. Orchestration refers to the coordination and execution of various security processes, ensuring they work together seamlessly. Automation automates routine and repetitive tasks, allowing for quicker incident handling and reducing the burden on human analysts. Response involves the implementation of predefined actions and workflows in response to security incidents, enabling a swift and consistent reaction to threats. One of the distinctive features of our SOAR platform is its automatic remediation capability. SOAR platforms not only enhance the efficiency of security operations but also enable better decision-making by providing analysts with relevant information and context. They play a pivotal role in strengthening a company's overall cybersecurity posture by enabling a proactive approach to identifying and mitigating potential threats.

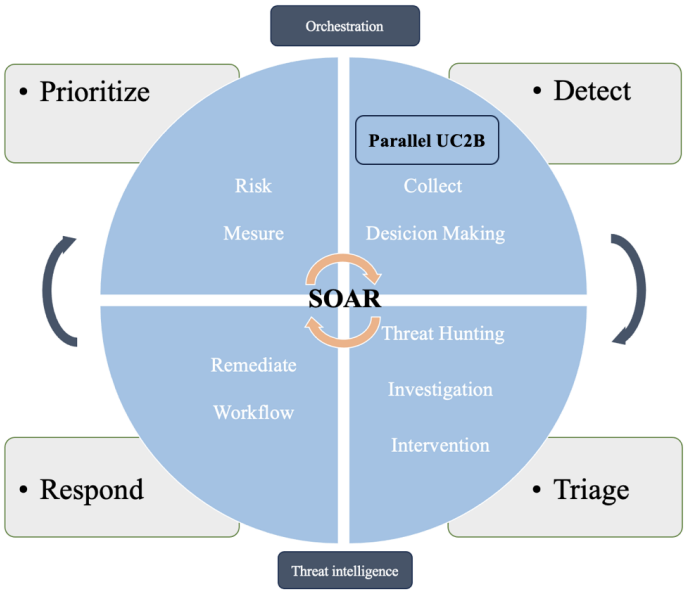

Fig. 2. Elements of Security Orchestration, Automation and Response

The surge in security breaches, both in frequency and sophistication, has significantly complicated the task of discerning genuine alerts from false positives, straining the resources of Security Operation Centers (SOCs). Currently, SOCs grapple with a deluge of alerts, with the majority turning out to be false alarms. This underscores the pressing need for more efficient detection and response mechanisms. Thus, the focus of this initiative lies in validating the effectiveness of the parallel UC2B approach and seamlessly integrating it into an existing SOAR platform. The ultimate aim is to elevate the system's detection capabilities, leading to higher accuracy rates and swifter responses to security incidents. This endeavor is a meaningful step towards bolstering cybersecurity defenses in light of the constantly changing threat landscape.

III. Experiments and their analysis

In this section, we will present the outcomes of our work conducted on Ruche Cluster. The evaluation of our approach was performed based on their respective architectures. Specifically, we implemented our algorithm on foor and eight nodes, corresponding to the number of parallel co-methods used, with

each node configured with 10 tasks considering the number of bags employed. To establish the connection between both hardware architectures and our implementation settings, we will begin by describing the hardware architecture of Ruche. Following that, we will provide an in-depth exploration of the obtained results.

A. Ruche Cluster: Single Node Specifications

The Ruche cluster, located at the Moulon mesocentre [2], is equipped with an Intel Xeon Gold 6230 CPU, based on the x86 Cascade Lake architecture. Each node boasts a total of 40 cores, distributed across 2 CPUs, with 20 cores per CPU. The processor operates at a base frequency of 2.10 GHz. In terms of cache, each core has 32 KB for instructions and 32 KB for data in L1, 1 MB in L2, and a shared L3 cache of up to 27.5 MB. Additionally, Ruche supports SIMD extensions up to AVX-512. It is comprised of a total of 216 nodes.

B. Results and Insights

In the following section, we present the results of our experiments, where we have chosen accuracy as the performance metric, given its appropriateness. Accuracy is a metric commonly used in classification tasks to measure the proportion of correctly classified instances out of the total instances in a dataset. It provides an indication of the model's effectiveness in making correct predictions across all classes and is a valuable measure for evaluating the overall performance of a classification algorithm.

The experiments have two primary objectives. Firstly, they aim to enhance the model's validation, which previously achieved an accuracy of 98% in a prior work. This improvement is pursued by increasing the number of parallel co-methods used and introducing variations. Secondly, the experiments seek to deliver a comprehensive demonstration of the model's performance capabilities.

The first experiment involves collaborating four parallel co-methods across four nodes, where each co-method is trained on ten parallel bags in turn.

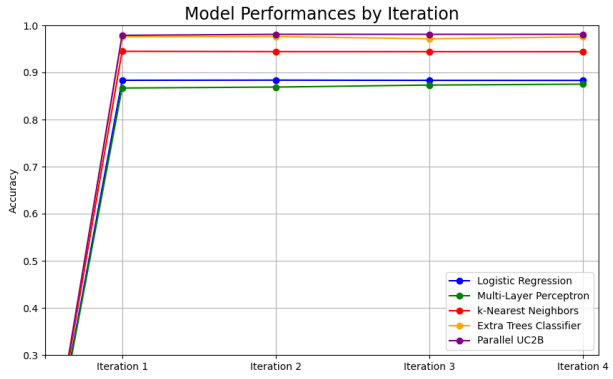

Fig. 3. Accuracies of parallel UC2B and 4 co-methods across the iterations

In Figure 3, we can discern the accuracy scores for various models over four iterations. Notably, in the initial iteration, LR achieves an accuracy of approximately 88.34%, while MLP attains around 86.69%. On the other hand, the kNN model demonstrates a significantly higher accuracy of about 94.49%. Moreover, ETC excels with an accuracy of approximately 97.58%. Remarkably, the ensemble model parallel UC2B consistently surpasses individual models, boasting an accuracy of approximately 97.88% in the first iteration and stabilizing at 98% from the second iteration onward. This prompts the question of whether the enhanced performance of UC2B over the ensemble will persist with the inclusion of more methods, or if it may encounter limitations.

This experiment involves collaborating six parallel co-methods across six nodes, where each co-method is trained on ten parallel bags in turn.

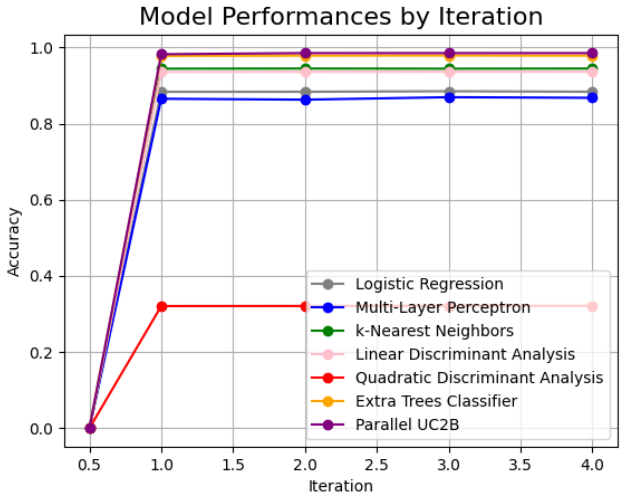

Fig. 4. Accuracies of parallel UC2B and 6 co-methods across the iterations

Figure 4 illustrates the results with six co-methods, providing insights into the performance of various models across four iterations. The LR model exhibits a modest improvement, advancing from 88.33% to 88.49% in accuracy over the iterations, while the MLP shows slight fluctuations within the range of 86.27% to 86.92%. The kNN model maintains a steady accuracy of about 94.40% throughout. Both LDA and QDA consistently maintain accuracies of 93.52% and 32.09% respectively. The ETC starts impressively at 97.77% and sustains a high accuracy of over 97.8% in subsequent iterations. Notably, parallel UC2B outperforms individual models with an accuracy of 98.2% in the first iteration, further improving to 98.5% in subsequent iterations, underscoring the potency of ensemble models in enhancing predictive accuracy.

This experiment involves collaborating eight parallel co-methods across eight nodes, where each co-method is trained on ten parallel bags in turn.

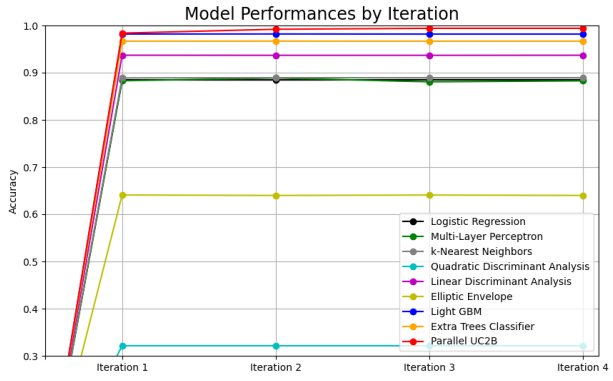

Fig. 5. Accuracies of parallel UC2B and 8 co-methods across the iterations

The graph 5 illustrates the performance evolution of eight distinct machine learning models across four iterations of UC process. These models encompass a diverse range of techniques, from LR to more complex ensemble methods. Among them, LDA demonstrates remarkably high accuracy, stabilizing at around 93.7%. Notably, LGBM and ETC exhibit outstanding performance, achieving accuracies of approximately 98.1-98.2% and 96.7-96.9% respectively. In contrast, QDA consistently struggles with a low accuracy of about 32.1%. However, the standout performer is the parallel UC2B model, which showcases an exceptional accuracy range of 98.4-99.4%. This model achieves this high accuracy by intelligently combining the strengths of all the co-methods and strategically reevaluating instances with false positives/negatives over multiple iterations. This approach demonstrates remarkable learning capability, continuously improving its performance with each iteration. It outperforms all other models, underscoring its exceptional effectiveness in the given task. These results offer valuable insights into the relative strengths and weaknesses of each model in this specific context.

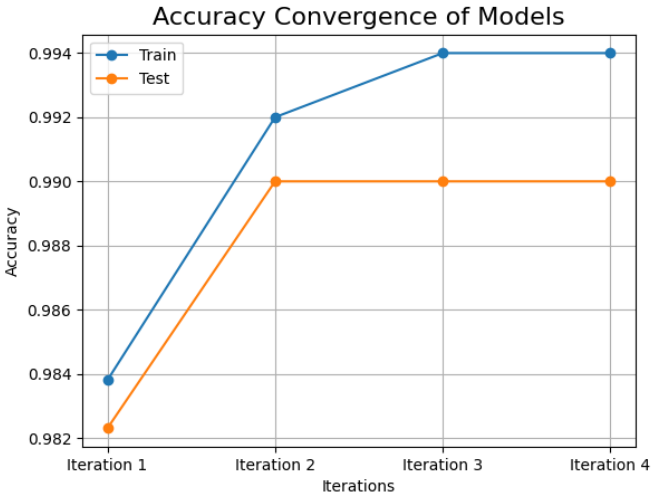

Fig. 6. Train and test accuracies of parallel UC2B using 8 co-methods.

The figure vividly demonstrates the convergence of parallel UC2B accuracy through four iterative processes of UC. The x-axis represents the number of times the training process was reiterated with restarted conditions, involving the reintroduction of FP/FN. On the y-axis, accuracy is graphically portrayed, quantifying the proportion of correctly classified instances. The training accuracy signifies how adeptly the models adapt to the training data in each iteration. Simultaneously, the testing accuracy elucidates the models proficiency in generalizing to new, unseen data. Initially, both training and testing accuracies are commendably high at approximately 98% and 98.2% respectively, indicating a robust initial performance. Subsequently, with each iteration of unite and conquer, a discernible upward trend is observed in both training and testing accuracies, indicating a consistent process of learning and refinement. By the fourth iteration, a steady convergence emerges, with both training and testing accuracies stabilizing around 99%. This compellingly suggests that the models have not only effectively absorbed knowledge from the data but have also consistently maintained a high level of accuracy.

The following graph serves to highlight that these experiments were conducted to assess the performance of the ensemble approach parallel UC2B using different and varying sets of co-methods. This allowed for a comprehensive analysis of their impact on the results.

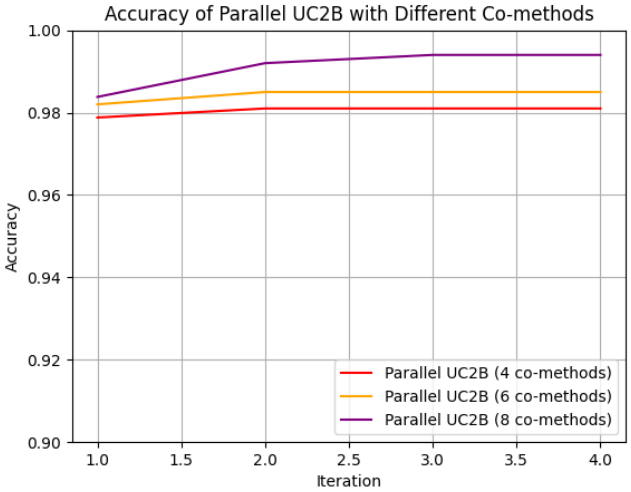

Fig. 7. Improving accuracy of parallel UC2B by adding co-methods

This figure, 7, demonstrates the impact of the number of co-methods on the improvement of parallel UC2B's accuracy across iterations. Beginning with four co-methods, parallel UC2B achieves an accuracy of 97.88% in the first iteration, steadily maintaining this level in subsequent iterations. When six co-methods are incorporated, a noticeable enhancement is observed, with the accuracy rising to 98.2% in the first iteration and further increasing to 98.5% in subsequent iterations. This trend continues as eight co-methods are utilized, leading to a substantial boost in accuracy. In the third and

fourth iterations, the model achieves an impressive accuracy of 99.4%. This highlights a positive correlation between the number of co-methods integrated and the accuracy enhancement of parallel UC2B, underlining the effectiveness of this ensemble approach in leveraging diverse models for improved predictive performance.

To assess the performance of parallel UC2B, we executed the implementation on the Ruche Cluster employing 8 nodes. The algorithm's parallelism is expressed using the Python language along with the mpi4py library. Performance is assessed in terms of the execution time observed with increasing dataset sizes. This metric stands as a fundamental gauge of performance scalability [1], demonstrating how execution time evolves with increasing data.

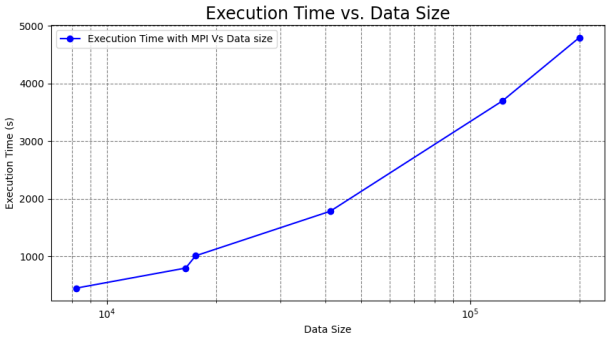

Fig. 8. Scaling performance of parallel UC2B with varying data sizes

The graph 8 illustrates the scaling performance of parallel UC2B in response to different data sizes. It showcases a positive correlation between execution time and database size, highlighting the benefits of the parallel UC2B approach. It's important to note that this analysis is based on a specific configuration employing 8 parallel co-methods distributed across 8 nodes. While a clear upward trend is observed, signifying that execution time increases with larger datasets, it's crucial to acknowledge that this increase occurs at a gradually diminishing rate. This suggests that the efficiency of parallelization may have a more pronounced impact for smaller datasets. These insights underscore the critical role of considering data size in the implementation of parallelization to optimize the performance of the UC2B framework. However, conducting a more extensive analysis to evaluate the framework's performance with even larger databases is a potential avenue for future research.

Given that augmenting the number of co-methods hasn't affected the efficiency of parallel UC2B, but has instead enhanced its accuracy. Moreover, increasing the database size without altering the available resources hasn't exhibited any scalability concerns. We intend to assess its performance in relation to execution time concerning both the augmentation of co-methods and the resources employed.

Fig. 9. Correlation between co-method count and execution time

The presented figure 9 unveils a notable trend, as the number of co-methods increases, there is a noticeable surge in execution time. This observation sheds light on potential limitations within the parallel processing framework. It is crucial to note that within a set of co-methods, there exists inherent variability in processing speeds, resulting in wait times at synchronization points. This phenomenon is amplified with a higher number of co-methods. Furthermore, communication between these co-methods scales linearly with their quantity, further contributing to extended execution times. Therefore, while it might be anticipated that an increase in co-methods would lead to enhanced efficiency, factors such as resource contention and management overhead play a substantial role in system performance. This emphasizes the need for careful evaluation and precise adjustment of parallel processing configurations to optimize resource utilization and minimize execution times. Ultimately, these findings underscore the need for a balanced consideration between accuracy improvement and temporal efficiency. They underscore that augmenting the number of co-methods may not always be advisable, as it is imperative to ensure that the gains in accuracy achieved do not disproportionately extend execution times.

These experiments have improved the validation of the parallel UC2B framework in detecting threats, which maintains good accuracy with a diverse and large number of machine learning models. Notably, with only 4 iterations of UC, symbolizing the re-injection of FP/FN into the training set for initiating new cycles, each model undergoes separate training bags over 10 iterations. Data scalability has also shown good performance. However, the significant increase in execution time, even with increased resources, could pose a limitation to this approach. This calls for further investigation and underscores the importance of adopting asynchronous communications between co-methods, as exemplified by the implementation of the asynchronous version of parallel UC2B without synchronization points, with larger datasets in future endeavors. Besides, this work contributes to the advancement of prior efforts, notably improving execution time. In comparison to UCEL, it achieves a noteworthy enhancement in the detection rate, elevating it from 97% on unseen data to 99%.

IV. Related Work

The state-of-the-art in anomaly detection within the field of cybersecurity has been advancing rapidly in recent years. Numerous studies and approaches have been proposed to address the challenge of detecting unusual and potentially harmful behavior in computer systems and networks. Some machine learning-based techniques applied to anomaly detection, including Bagging (which involves training multiple models on different data subsets and combining their predictions) and Boosting methods (that improve a model's accuracy by emphasizing misclassified examples [6]), run alongside spectral calculations [18] that involve analyzing eigenvalue and eigenvector values.

More recently, Diop et al. applied the Unite and Conquer approach [12] used in linear algebra to ensemble learning. The resulting technique, called UCEL, iteratively boosts a set of methods that work like bagging, and iterations of this boosting continue until the desired accuracy is achieved [9], [10]. This extended method shows improved performance.

Moreover, there have been significant efforts in evaluating these methods and comparing their performance on various data sets, including the widely recognized UNSW-NB15 data set [20]. The UNSW-NB15 data set, with its large number of simulated network traffic instances, is commonly used for evaluating the performance of anomaly detection algorithms in a realistic setting. It contains a wide range of attack types and is characterized by its high volume and high dimensionality, making it a challenging data set for anomaly detection algorithms.

In addition to the previously mentioned Bagging and Boosting methods and spectral calculations, other notable methods include Variational Autoencoders (VAE), which learn a probabilistic representation of normal data and identify anomalies based on the reconstruction probability [4]. Generative Adversarial Networks (GAN) have been applied to anomaly detection, where a generator reproduces normal data and a discriminator distinguishes between real and generated data [3]. Hidden Markov Models (HMM) have been employed for anomaly detection, extending the one-class support vector machine (SVM), by leveraging latent dependency structures [14]. The approach achieves superior anomaly detection performance compared to traditional one-class SVM, as demonstrated through empirical evaluations on diverse datasets in computational biology and computational sustainability domains. Recurrent Neural Networks (RNN), such as LSTM, have been effective in capturing sequential dependencies for anomaly detection in time series data [19]. These methods, along with preprocessing techniques for feature selection and data normalization, have contributed to the advancement of anomaly detection in cybersecurity.

As the application of anomaly detection techniques expands beyond the cybersecurity domain, researchers are actively exploring their adaptability to various specific application fields. This progression is exemplified by recent studies proposing innovative approaches to address real-time monitoring challenges in complex systems.

To solve the problem of real-time monitoring of the signals produced by the accelerators, a fault detection method is proposed in [16]. This method, based on data from the beam position monitoring system, can identify anomalies in SLAC's radio frequency (RF) stations and detect more events while reducing false positives compared to diagnostics of existing RF stations.

Moreover, the method CoAD proposed in [17], trains anomaly detection models on unlabeled data, based on the expectation that anomalous behavior in one sub-system will produce coincident anomalies in downstream sub-systems.

Furthermore, The lack of structured parallel implementation in anomaly detection poses a significant challenge for the field [13]. Anomaly detection algorithms often involve complex computations and deal with large datasets, making them computationally demanding. While parallel computing has the potential to accelerate these tasks by distributing the workload across multiple processing units, achieving efficient parallel implementations is not straightforward [7], [21]. Many anomaly detection methods are not inherently parallelizable due to their sequential nature and data dependencies, requiring substantial modifications for parallel processing. Load imbalance among processing units, caused by the irregularity of anomaly occurrence in data, further complicates the parallelization process. Additionally, the absence of standardized parallel frameworks tailored explicitly for anomaly detection hinders progress [11]. To address these issues, focused research, collaboration between anomaly detection and parallel computing experts, and the development of specialized parallel frameworks are essential to unlock the benefits of parallel computing in advancing anomaly detection capabilities.

In this paper, we have introduced the Scalable Anomaly Detection with UC2B framework, which harnesses computational resources for scalable anomaly detection in cybersecurity. We delved into the feedback garnered from experiments conducted on Ruche Cluster, affirming the effectiveness of the framework. While this scalable framework is versatile and applicable across various domains, our core focus remains on the detection of diverse cybersecurity threats, placing special emphasis on analyzing the UNSW-NB15 dataset. Notably, by incorporating the parallel UC2B extension, we have introduced a layer of multi-level parallelism to the UC2B framework, significantly enhancing its processing efficiency. Through this work, we aim to propel the evolution of anomaly detection methods, bolster the defense against emerging cyber threats, and proficiently address abnormal behaviors.

V. Conclusion & Perspectives

In this endeavor, the aim is to seamlessly integrate parallel UC2B into an operational SOAR platform, enhancing its practical utility in cybersecurity. This integration strengthens the platform's ability to identify anomalies and security threats, marking a significant advancement in network security. The paper focuses on improving the detection rate of parallel

UC2B by introducing collaborative co-methods and evaluating their impact on execution time and scalability.

The process begins with continuous data collection from diverse network sources, followed by preprocessing step. Trained classifiers within each co-method actively monitor network activities, identifying deviations from established norms. Upon detecting anomalies, alerts prompt thorough investigations by analysts, who prioritize and address them while meticulously documenting their actions. The integrated system adapts to evolving threats through ongoing assessment and refinement, solidifying its pivotal role in network security. Additionally, it showcases robust multi-level parallelism attributes, such as fault tolerance, adaptability to diverse architectures, and proficient load balancing capabilities, confirming its suitability for real-world applications in cybersecurity.

Our investigation on the Ruche cluster involved intricate experimentation, assessing both model and data parallelism for various machine learning co-methods. The evaluation concentrated on LR, MLP, KNN, EE, QDA, LDA, LGBM, and ETC models within the parallelization framework, identifying parallel UC2B as a robust and accurate approach. The study highlights the model's efficiency and scalability with consistent execution times across larger datasets. A comparison with UC2B and UCEL highlights the efficiency gained through modern supercomputers and advanced parallelization.

Looking ahead, our focus centers on enhancing the parallel UC2B framework with asynchronous communication capabilities to reduce execution time and enhance accuracy. We also aim to incorporate potent neural network-based co-methods to further heighten the detection rate. This upgrade will be put to the test on extensive datasets and a larger number of computing nodes. Our overarching objective remains the development of a versatile framework for efficient anomaly detection across diverse application domains and datasets of varying sizes.

ACKNOWLEDGMENT

The previous work UC2B and this research was performed using HPC resources from the "Mésocentre" computing center of CentraleSupélec, École Normale Supérieure Paris-Saclay and Université Paris-Saclay supported by CNRS and Région Île-de-France (https://mesocentre.universite-paris-saclay.fr/). We sincerely thank Research engineer Martial Mancip for his kind assistance, which greatly aided our study.

REFERENCES

[1] Database scalability — Wikipedia, the free encyclopedia. [Online; accessed 14-August-2023].

[2] Ruche cluster of mesocentre, October 1 2020.

[3] Samet AKCAY, Amir ATAPOUR-ABARGHOUEI et Toby P BRECKON : Ganomaly: Semi-supervised anomaly detection via adversarial training. *In Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 622–637. Springer, 2019.

[4] Jinwon AN et Sungzoon CHO : Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015.

[5] Archana ANANDAKRISHNAN, Senthil KUMAR, Alexander STATNIKOV, Tanveer FARUQUIE et Di XU : Anomaly detection in finance: editors' introduction. *In KDD 2017 Workshop on Anomaly Detection in Finance*, pages 1–7. PMLR, 2018.

[6] Owais BUKHARI, Parul AGARWAL, Deepika KOUNDAL et Sherin ZAFAR : Anomaly detection using ensemble techniques for boosting the security of intrusion detection system. *Procedia Computer Science*, 218:1003–1013, 01 2023.

[7] Franck CAPPELLO, Al GEIST, William GROPP, Sanjay KALE, Bill KRAMER et Marc SNIR : Toward exascale resilience: 2014 update. *Supercomputing Frontiers and Innovations*, 1(1):5Äì28, Jun. 2014.

[8] Varun CHANDOLA, Arindam BANERJEE et Vipin KUMAR : Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[9] Abdoulaye DIOP, Nahid EMAD et Thierry WINTER : A parallel and scalable framework for insider threat detection. *In 27th IEEE International Conference on High Performance Computing, Data, and Analytics, HiPC 2020, Pune, India, December 16-19, 2020*, pages 101–110. IEEE, 2020.

[10] Abdoulaye DIOP, Nahid EMAD et Thierry WINTER : A unite and conquer based ensemble learning method for user behavior modeling. *In 39th IEEE International Performance Computing and Communications Conference, IPCCC 2020, Austin, TX, USA, November 6-8, 2020*, pages 1–8. IEEE, 2020.

[11] Qian DU, Bo TANG, Weiying XIE et Wei LI : Parallel and distributed computing for anomaly detection from hyperspectral remote sensing imagery. *Proceedings of the IEEE*, 109(8):1306–1319, 2021.

[12] Nahid EMAD et Serge G. PETITON : Unite and conquer approach for high scale numerical computing. *J. Comput. Sci.*, 14:5–14, 2016.

[13] Siavash GHIASVAND et Florina M. CIORBA : Anomaly detection in high performance computers: A vicinity perspective. *In 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC)*, pages 112–120, 2019.

[14] Nico GÖRNITZ, Mikio BRAUN et Marius KLOFT : Hidden markov anomaly detection. *In International conference on machine learning*, pages 1833–1842. PMLR, 2015.

[15] Cheng-Yuan HO, Yuan-Cheng LAI, I-Wei CHEN, Fu-Yu WANG et Wei-Hsuan TAI : Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems. *IEEE Communications Magazine*, 50(3):146–154, 2012.

[16] Ryan HUMBLE, Finn H. O'SHEA, William COLOCHO, Matt GIBBS, Helen CHAFFEE, Eric DARVE et Daniel RATNER : Beam-based rf station fault identification at the slac linac coherent light source. *Phys. Rev. Accel. Beams*, 25:122804, Dec 2022.

[17] Ryan HUMBLE, Zhe ZHANG, Finn O'SHEA, Eric DARVE et Daniel RATNER : Coincident learning for unsupervised anomaly detection, 2023.

[18] Tomilayo KOMOLAFE, A Valeria QUEVEDO, Srijan SENGUPTA et William H WOODALL : Statistical evaluation of spectral methods for anomaly detection in static networks. *Network Science*, 7(2):238–267, 2019.

[19] Pankaj MALHOTRA, Lovekesh VIG, Gautam SHROFF, Puneet AGARWAL *et al.* : Long short term memory networks for anomaly detection in time series. *In Esann*, volume 2015, page 89, 2015.

[20] Nour MOUSTAFA et Jill SLAY : The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. pages 1–14, 01 2016.

[21] Shashank SHANBHAG et Tilman WOLF : Accurate anomaly detection through parallelism. *IEEE Network*, 23(1):22–28, 2009.

[22] Ljiljana STOJANOVIC, Marko DINIC, Nenad STOJANOVIC et Aleksandar STOJADINOVIC : Big-data-driven anomaly detection in industry (4.0): An approach and a case study. *In 2016 IEEE international conference on big data (big data)*, pages 1647–1652. IEEE, 2016.

[23] Chee-Wooi TEN, Junho HONG et Chen-Ching LIU : Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*, 2(4):865–873, 2011.

[24] Stephane TUFFERY : *Processing of Large Volumes of Data*, pages 49–70. 2023.

[25] Arijit UKIL, Soma BANDYOAPDHYAY, Chetanya PURI et Arpan PAL : Iot healthcare analytics: The importance of anomaly detection. *In 2016 IEEE 30th international conference on advanced information networking and applications (AINA)*, pages 994–997. IEEE, 2016.

[26] Rahul VAST, Shruti SAWANT, Aishwarya THORBOLE et Vishal BADGUJAR : Artificial intelligence based security orchestration, automation and response system. *In 2021 6th International Conference for Convergence in Technology (I2CT)*, pages 1–5, 2021.