Enhancing the Parallel UC2B Framework: Approach Validation and Scalability Study

Zineb Ziani[1,2][0009−0004−2095−2911], Nahid Emad[1], Miwako Tsuji[3], and Mitsuhisa Sato[3]

[1] University of Paris Saclay / Li-PaRAD / MDLS, Paris, France
[2] Numeryx, 17 Rue Jeanne Braconnier, 92360 Meudon, France
[3] RIKEN Center for Computational Science, Kobe, Hyogo 650-0047, Japan

Abstract. Anomaly detection is a critical aspect of uncovering unusual patterns in data analysis. This involves distinguishing between normal patterns and abnormal ones, which inherently involves uncertainty. This paper presents an enhanced version of the parallel UC2B framework for anomaly detection, previously introduced in a different context. In this work, we present an extension of the framework and present its large-scale evaluation on the Supercomputer Fugaku. The focus is on assessing its scalability by leveraging a great number of nodes to process large-scale datasets within the cybersecurity domain, using the UNSW-NB15 dataset. The ensemble learning techniques and inherent parallelizability of the Unite and Conquer approach are highlighted as key components, contributing to the framework's computational efficiency, scalability, and accuracy. This study expands upon the framework's capabilities and emphasizes its potential integration into an existing Security Orchestration, Automation, and Response (SOAR) system for enhancing cyber threat detection and response.

Keywords: Anomaly Detection · Linear Algebra · Unite and Conquer Approach · Machine Learning · High performance computing · Ensemble learning · Uncertainties · UC2B · UCEL · Cybersecurity.

1 Introduction

Anomaly detection is a crucial element in data analysis and has gained widespread recognition for its ability to identify patterns or behaviors significantly deviating from normal or expected observations, with diverse applications across various domains [7].

In the field of finance, anomaly detection plays a pivotal role in detecting fraud and suspicious financial activities [4]. By identifying unusual transactions, atypical spending patterns, or fraudulent behaviors, anomaly detection contributes to fortifying the security and protection of financial assets. Likewise, in manufacturing, it is employed to monitor production processes, identifying failures or unexpected variations to enhance product quality, optimize operations, and minimize downtime [24]. Additionally, in healthcare, anomaly detection aids

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63783-4_26

in detecting unusual symptoms, identifying rare diseases, and analyzing medical images for precise diagnoses and timely interventions [29].

In cybersecurity, anomaly detection serves as a vital tool for safeguarding computer systems against malicious attacks [26]. It plays a crucial role in real-time threat detection and prevention by identifying abnormal behaviors on networks, data breaches, hacking activities, and intrusion attempts, thereby enhancing system security and minimizing the impact of cyber threats.

Despite advancements in anomaly detection driven by large-scale datasets and sophisticated machine learning algorithms, challenges persist due to the increasing complexity and size of modern datasets [12]. Efficient processing and analysis of data require substantial computational power, with deep learning models relying on high-performance GPUs or specialized hardware accelerators for training and fine-tuning. Real-time anomaly detection, on the other hand, demands rapid analysis of incoming data streams, necessitating the processing speed and scalability of modern hardware. Organizations must invest in powerful computational resources to fully harness the potential of these advanced techniques for effective anomaly detection in complex data.

To address these challenges, we have developed the parallel UC2B Framework in [30], an acronym for Unite and Conquer with Bagging and Boosting. Unite and Conquer is an iterative method that involves making several iterative methods collaborate by sharing their information. Bagging involves training multiple models on different data subsets and combining their predictions, and boosting improves a model's accuracy by emphasizing misclassified examples [25]. In previous experiments utilizing the parallel UC2B framework for anomaly detection, specifically on the smallest data set of UNSW-NB15, notable efficiency was demonstrated, achieving a detection rate ranging from 97% to 99% [30]. This framework leverages the Unite and Conquer methodology, integrating Bagging and Boosting techniques to primarily enhance prediction accuracy. However, these experiments have brought to light scalability concerns, particularly in managing the computational demands of extensive datasets. The number of nodes is confined by the number of co-methods, as each co-method necessitates a dedicated node for training. This interdependency introduces inefficiencies, affecting both memory usage and computational complexity. Addressing these challenges is crucial for the seamless implementation of the synchronous version of Parallel UC2B in real-world anomaly detection solutions.

In this paper, we introduce an enhanced version of the UC2B framework, a parallel anomaly detection system designed for cybersecurity threat detection. Conducting experiments on the Supercomputer Fugaku with up to 40 nodes, our versatile framework adeptly addresses diverse cybersecurity threats, with a specific emphasis on analyzing the biggest dataset of UNSW-NB15, which comprises over 2.5 million samples [18]. The Parallel UC2B extension integrates multi-level parallelism and double bagging, significantly enhancing processing efficiency. Our objectives encompass advancing the framework, fortifying defenses against emerging cyber threats, and facilitating its integration into existing SOAR systems. We validate its high accuracy, assess its effectiveness through confidence

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63783-4_26

score calculations, and study its scalability under both weak and strong loads, including its behavior with larger databases. Key elements of our work include:

- Enhanced exploration of an optimized configuration incorporating multi-level parallelism, a fusion of double bagging and boosting, complemented by a restarting strategy inspired by the unite and conquer method for anomaly detection.
- Integration of a diverse array of components, encompassing various ML models, with inherent load balancing potential, distributed computation capabilities, and a fault-tolerant implementation strategy.
- Adoption of both model parallelism and data parallelism.
- Implementation of a parallel framework designed to harness the performance capabilities of high-performance computing architectures.
- Validation of the framework's efficacy through a series of experiments executed on the supercomputer Fugaku using 40 nodes.
- Specialized focus on the application of the framework within the cybersecurity domain, leveraging the UNSW-NB15 dataset for evaluation.
- Integrating a robust uncertainty metric deepens predictive insights, bolstering confidence in model performance and decision-making.

2 State of the art

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 [5]), run alongside spectral calculations [16] that involve analyzing eigenvalue and eigenvector values.

More recently, Diop et al. applied the Unite and Conquer approach [11] 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 [8, 9]. This extended method shows improved performance. Another combination of these techniques was presented in the article [30] to improve the results of UCEL in terms of detection rate.

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 [18]. 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.

DOI: 10.1007/978-3-031-63783-4_26

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 [3]. 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 [2]. Hidden Markov Models (HMM) have been employed for anomaly detection, extending the one-class support vector machine (SVM), by leveraging latent dependency structures [13]. 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 [17]. 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 [14]. 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 [15], 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 [12]. 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 [6, 23]. 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 [10]. 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.

3 Software Architectures of Enhanced Parallel UC2B

In various scientific disciplines, the escalating data generation surpasses computational capacities, compelling the integration of modeling, analysis, and high-performance computing [21]. These challenges, spanning diverse fields, are rooted in applied mathematics, including linear algebra and statistics, alongside artificial intelligence, which encompasses machine learning methods and high-performance computing techniques. Within cybersecurity, the evolving subtlety of security breaches extends investigation times, demanding a discerning approach to distinguish authentic alerts from false alarms. Expertise and timely validation of 'false alerts' are crucial in a Security Operations Center (SOC) [20]. 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, is another anomaly detection technique proposed by Diop et al. in [8], [9], called UCEL. In this paper, we propose an enhanced version of UCEL which improves its performance. To distinguish this extension from UCEL, we call it Parallel 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, in addition to a double bagging. 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.

3.1 Unite and Conquer Approach

"Unite and Conquer" is a problem-solving paradigm that orchestrates multiple iterative methods, or co-methods, to collectively address complex problems, particularly in linear algebra [11]. Applied in resolving expansive, sparsely populated linear systems and eigenvalue predicaments, this approach accelerates convergence by aggregating intermediate outcomes from each co-method. The strategic restarting approach plays a pivotal role in providing a better starting point for each new cycle, enhancing overall convergence. Co-methods exchange intermediate solutions to determine effective restarting conditions, resulting in swifter global convergence. With intrinsic advantages like multi-level parallelism, robust fault tolerance, adaptability to component heterogeneity, asynchronous communication capabilities, and inherent load balancing potential, the "Unite and Conquer" approach is well-suited for cutting-edge computational architectures. It optimally allocates computational resources, enhancing efficiency and parallel processing benefits, accelerating problem resolution, and maximizing resource utilization.

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

DOI: 10.1007/978-3-031-63783-4_26

$k = 0$) 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 intermediate 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 $\ell$ **do in parallel**
  Compute $S_i^k$ by applying $L_i$ to $P$ with initial condition$I_i^k$.
  If $S_i^k$ is sufficiently accurate, STOP all $\ell$ 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 $k$.
```

Essentially, this approach boasts a simple yet versatile framework applicable to various iterative methods, as exemplified in this paper. We specifically explore integrating boosting techniques within bagging methodologies, introducing a second level of parallelism to enhance the model's adaptability and performance across diverse datasets and scenarios.

3.2 Parallel UC2B Insights

The Parallel UC2B framework aims to enhance anomaly detection accuracy and efficiency by integrating the Unite and Conquer problem-solving approach with Bagging, Boosting, and multi-level parallelism. The objective is to iteratively improve accuracy, ensure high confidence scores, and expedite anomaly identification. Collaboration among parallel co-methods refines their performance through multiple training cycles, culminating in a convergence state with substantial and stable improvements. Each co-method undergoes parallel training in inner bags of the dataset, emphasizing a multi-level parallelism approach.

In light of the constraint that LM models in scikit-learn cannot be trained on multiple nodes, we adopt a double bagging approach. This involves partitioning the database into outer bags through 'Node-based dataset partitioning' (cf. Fig. 1), ensuring the number of outer bags aligns with the total number of nodes divided by the number of co-methods (ML models). Subsequently, we thoroughly evaluate co-method performance using a validation set. Co-methods share their misclassified data (False Positives/Negatives), incorporating the boosting principle to adjust weights for misclassified samples during iterations based on co-method performance metrics. This iterative process heightens the likelihood of selecting crucial samples for constructing training data in subsequent cycles.

Resulting in inner bags of the original training data size from the boosted training dataset, this collaborative process allows each co-method to learn from its peers and gain insights into challenging data samples. The joint effort contributes to the gradual refinement and improvement of the models.

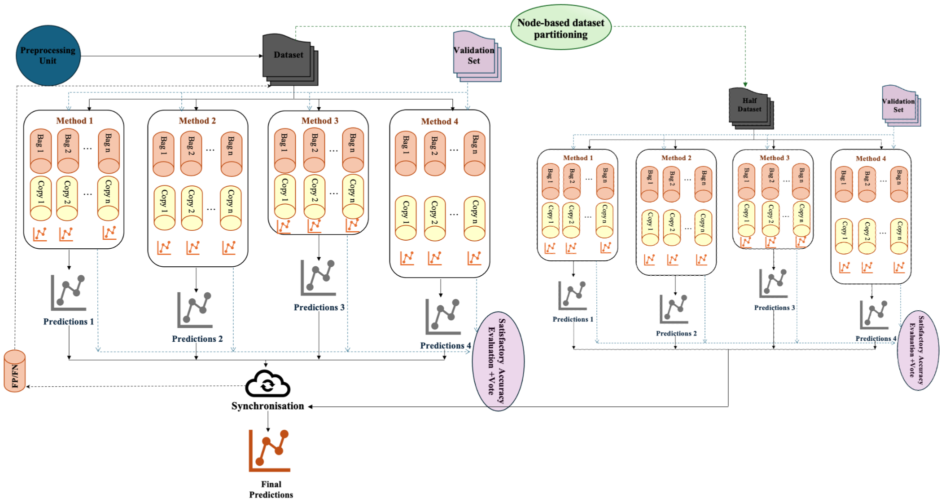


Fig. 1: Enhanced Parallel UC2B Architecture.

The goal of Parallel UC2B is versatility, addressing a wide spectrum of attacks, whether internal or external, and anomalies, while maintaining 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. Given that UC approach learns the underlying global structure of data, we provide the entire dataset to all methods in parallel. Each co-method then creates duplicates of itself (depicted by yellow cylinders in 1) and segments the dataset into multiple bags (illustrated by orange cylinders in 1), training each copy of the co-method on a bag. 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, updating the weights of misclassified instances (FP/FN chunk), and checking if satisfactory accuracy has been achieved by testing on the validation dataset (purple arrows). In contrast, in UCEL [9], 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, providing flexibility with ”n” bags for each method.

In Unite and Conquer, our focus is primarily on synchronous communications among co-methods, with asynchronous communications also accommodated. The collaborative mechanism integrates bagging and boosting techniques for diverse data treatment, effectively balancing bias mitigation and variance management. The training process incorporates feedback from all co-methods, addressing performance metrics and instances of FP/FN. Furthermore, inherent parallelism optimally utilizes computational resources, enhancing efficiency, especially in ’parallel UC2B,’ where thread Parallelism and SIMD operations drive concurrent task handling, data processing, and collaborative sharing among

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63783-4_26

co-methods. This streamlined approach, complemented by efficient data Input/Output, ensures timely information exchange, supports boosting mechanisms, and yields significant performance gains.

3.3 Algorithm of Enhanced Parallel UC2B

In the realm of machine learning, the choice of data analysis methods hinges on the nature of available information, whether it's labeled, unlabeled, or imbalanced. In corporate environments, routine activities prevail, leading to datasets predominantly skewed towards normal behavior. The prevalence of normal data introduces challenges for anomaly detection.

In this implementation, we employ Gaussian Naive Bayes (GNB), Isolation Forest (IForest), Decision Tree (DT), and Random Forest (RF) as co-methods in the Parallel UC2B framework. This collaborative approach addresses challenges faced by traditional supervised and unsupervised methods when dealing with limited abnormal examples. Unsupervised techniques, adept at handling imbalanced data, primarily focus on identifying deviations without delving into their underlying causes. In contrast, supervised methods excel in scenarios with balanced and labeled datasets, but achieving such balance is often impractical in real-world applications.

Algorithm 2 Enhanced Parallel UC2B

```
Input:
Data set $D$.
Number of bags $I$.
Number of all process iterations $n$.
Number of learners $M$.
Sample weights $W$ initialized to ones.
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 $y_k$.
        Calculate misclassification rates using Predictions and true labels.
        $\beta \leftarrow 1.1 \times \text{misclassified} + 0.9 \times \text{classified}$
    Sync and Share $\beta$ and the results with all other processes.
    Check for desired accuracy; if met, stop all processes and exit.
Restart by Updating the input data with adjusted weights for the next iteration:
$W = W \times \beta$
$D \leftarrow$ Updated $D$ with adjusted sample weights $W$.
Output:
Obtain the boosted predictions after the desired iterations.
```

Our proposed approach begins with meticulous pre-processing of a dataset containing more than 2.5 million samples. This includes crucial steps such as data cleaning, feature selection, as well as scaling and normalization procedures. Subsequently, from this refined dataset, distinct sets for training, validation, and testing are carefully curated.

Following the node-based dataset partitioning, each model is trained in parallel to the others in inner parallel bags. Based on the predictions obtained, the coefficient β is calculated using the formula $\beta = 1.1 \times \text{misclassified} + 0.9 \times \text{classified}$. This coefficient is then used to update the instances that were misclassified for a subsequent boosted iteration.

4 Experiments and Analysis

In this section, we present results from our experiments on the Supercomputer Fugaku, utilizing 40 nodes for assessment. We'll explore the Fugaku hardware specifics to align with our implementation settings, followed by validation in the first subsection and performance demonstration in the second.

Fugaku is a supercomputer that boasts a highly advanced hardware architecture, positioning it as the most powerful supercomputer in the world in 2020 and 2021 [27, 28], and it is currently ranked as the number 2 supercomputer in 2023 [1]. It incorporates a state-of-the-art hardware design aimed at delivering exceptional performance and efficiency [19]. At its core, Fugaku utilizes the A64FX processor [22], which is based on the ARM architecture. Each A64FX chip comprises 48 computing cores, and each core is equipped with two 512-bit wide SIMD units. Powered by the A64FX chip, incorporates high-bandwidth memory (HBM2) modules, delivering substantial capacity and impressive bandwidth. To facilitate swift data transfer and node communication, Fugaku utilizes the custom-designed Tofu-D interconnect system. This network, based on a 6-dimensional mesh/torus topology, ensures efficient and low-latency interactions between nodes, enabling seamless data exchange and synchronization during parallel computations.

4.1 Validation of the Approach

The goal of this validation is to showcase that the Parallel UC2B approach achieves high accuracy with a robust confidence score.

We initiate our experiments by displaying the accuracy obtained on the training set during the UC iterations. As a reminder, the UC iterations involve the re-injection of False Positives (FP) and False Negatives (FN), updating misclassified instances through the β factor. Subsequently, we plot the curve obtained on the test dataset, which has never been seen by the framework.

Accuracy is chosen for performance evaluation in our context as it represents correct predictions relative to the total sample count. As depicted in Figure 2, illustrating accuracies obtained in the training set, IForest emerges as the weakest among the co-methods. While GNB, Decision Tree, and Random Forest exhibit high accuracies, they lack stability. In contrast, Parallel UC2B demonstrates convergence over UC iterations, achieving the highest stable accuracy.

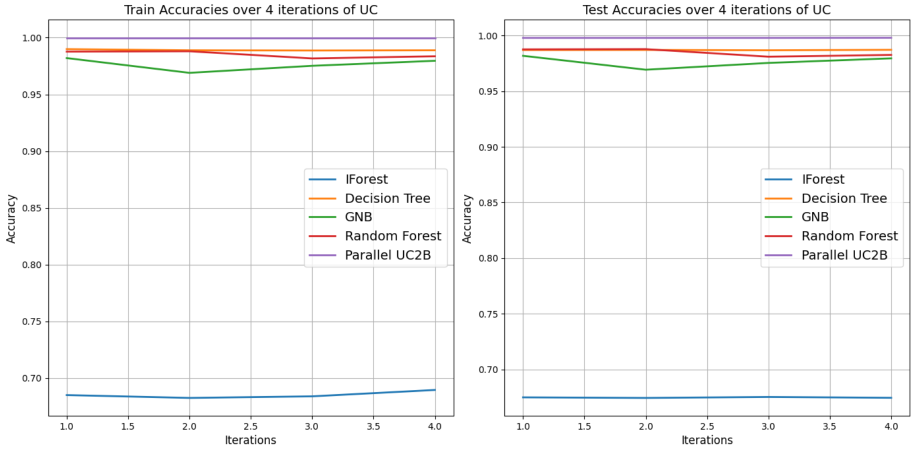


Fig. 2: Train & Test Accuracies: 4 Co-methods & Parallel UC2B (4 UC Iterations)

Transitioning to the test set graph, a similar pattern emerges. Other models display good accuracy but lack stability, while Parallel UC2B maintains its superiority in accuracy and stability even on data unseen by the framework.

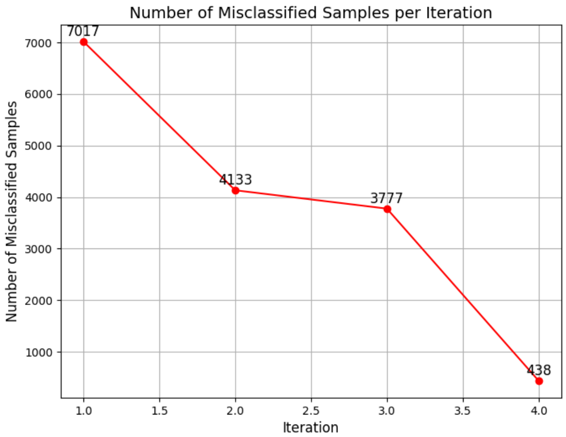


Fig. 3: The reduction of misclassified instances throughout the UC iterations.

This graph 3 illustrates the reduction of misclassified instances during the UC iterations on a node. As the data is divided based on nodes, this graph visually depicts the decrease in the number of misclassified instances, attributed

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63783-4_26

to the β factor updating these instances in each iteration.

Train Accuracy of Parallel UC2B	0.9999	0.9999	0.9999	0.9999
Test Accuracy of Parallel UC2B	0.9979	0.9979	0.9979	0.9980
Confidence Score of Parallel UC2B	0.9980	0.9980	0.9980	0.9981

Table 1: Score confidence of Parallel UC2B

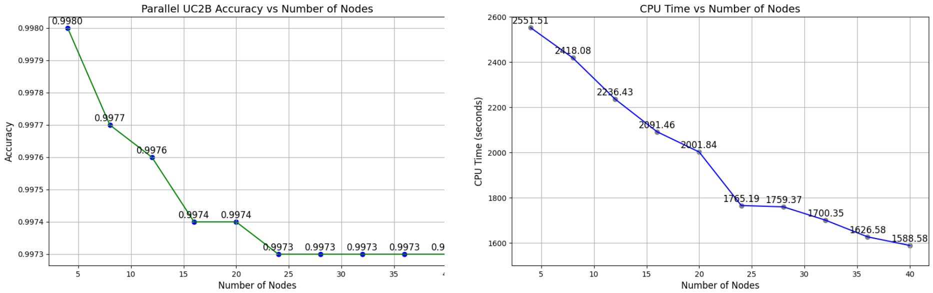
To measure the model's effectiveness, we employed the formula:

$$\textit{Confidence Score} = 1 - |\textit{Train Accuracy} - \textit{Test Accuracy}|$$

This score serves as a metric for assessing the model's consistency and stability across both training and test datasets. A higher Confidence Score, converging towards 1, indicates comparable performance on both datasets, highlighting the model's robustness. Conversely, a score closer to 0 suggests substantial differences between training and test data performances, potentially signaling instability or overfitting. In our specific case, the confidence scores for various iterations of the Parallel UC2B model consistently hover around 0.998. This steadfastness underscores a robust alignment between accuracy on training and test data, affirming the model's capability to generalize effectively to new data, which is a critical characteristic for model reliability.

4.2 Performance Demonstration of the Approach

After validating the accuracy of the approach, achieving a 99% accuracy on previously unseen test data and obtaining a very high confidence score, which improves upon the results obtained with the UCEL [8] and the previous version of Parallel UC2B [30], we will now focus on studying and evaluating the scalability of the framework, both strong and weak, along with its speedup and accuracy stability across different data sizes.



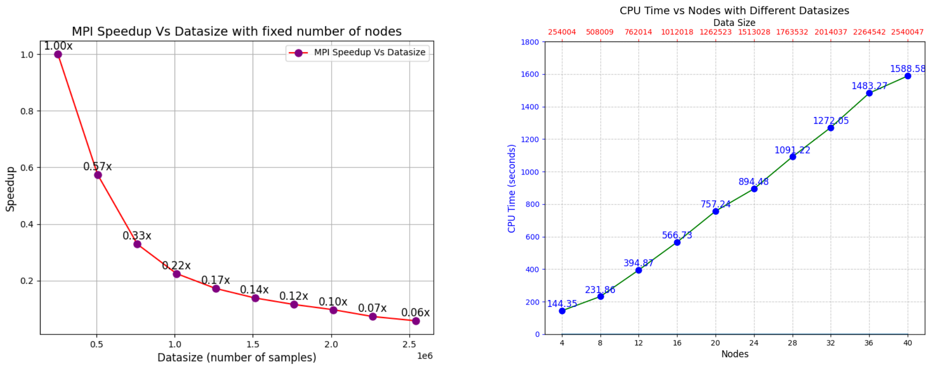
(a) Accuracy Stability

(b) Strong Scalability

Fig. 4: Accuracy Stability and Strong Scalability with Fixed Data-size

The graph 4a illustrates the accuracy achieved on a dataset comprising over 2.5 million samples as the number of nodes increases. Remarkably, the accuracy remains stable, showing no more than a 0.007% degradation from 4 to 24 nodes. Beyond 24 nodes, the accuracy plateaus at 0.9973, persisting even up to 40 nodes. This suggests that scaling the framework with additional nodes has negligible impact on accuracy.

As for graph 4b, it depicts the study of strong scalability, where we maintain a fixed problem size, increase the number of nodes, and evaluate speedup. In our case, the problem size exceeds 2.5 million, and even though the speed from 4 to 40 nodes doesn't quite double, it shows almost linear scalability up to 20 nodes. For node 24, we observe a speedup that deviates from linearity, followed by a near-linear recovery at node 28. The speed increase, while not doubling or more, can be attributed to synchronous communications between co-methods. Scaling from 4 to 40 nodes increases the number of communications tenfold. Additionally, the β factor, updating misclassified samples, is implemented in a way that each co-method needs to receive β for the entire 2.5 million data samples. This explains the suboptimal speedup obtained in this figure.



(a) Speedup Vs Data Sizes (12 Nodes)

(b) Weak Scalability

Fig. 5: Speedup with Fixed and Various Nodes and Various Data-sizes

The figure 5a illustrates the behavior of speedup using a fixed number of nodes, 12 in our case, while increasing the database size. We observe a degradation in speedup as the database size grows, which is expected given the constant number of nodes. However, starting from a database size of 1 million samples, we notice that the speedup does not degrade significantly. Interestingly, the execution time for 2.3 million samples is nearly the same as that for 2.6 million, indicating that the framework can effectively handle very large databases.

The graph in 5b illustrates weak scalability, which involves increasing the problem size in proportion to the addition of nodes. Ideally, for this experiment, the execution time would remain constant, as the increase in the database size

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63783-4_26

is accompanied by a corresponding increase in the number of nodes. However, due to the same phenomenon explained in strong scalability, when the number of nodes increases, the number of synchronous communications also increases with the growth of the database size, leading to an increase in the size of β. Nevertheless, the obtained curve is almost linear, indicating that the execution time is proportional to the addition of nodes and datasize.

In summary of the interpreted results, the framework stands out for its robust detection capability, marked by a high confidence score. Regarding its scalability performance, the framework exhibits remarkable adaptability to large-scale databases, maintaining stable accuracy even with an increased number of nodes. While strong scalability shows a proportional trend in some sections, weak scalability displays an almost linear trajectory. However, it is important to note an impact on speed performance. As the database expands, synchronous communications and the size of β increase, contributing to a gradual rise in execution time. This observation underscores the delicate balance between expanding computational resources and addressing challenges related to increased inter-node communication. The synchronous implementation of the framework is identified as the source of these observations. These findings highlight the importance of considering optimal system configurations for large-scale deployments, suggesting a possible solution in developing the asynchronous version of the framework.

5 Conclusion

This study presents a comprehensive exploration of the enhanced Parallel UC2B framework for anomaly detection, evaluated on the supercomputer Fugaku. The core analysis revolves around assessing IForest, GNB, DT, and RF models within the parallelization framework, with the Parallel UC2B model emerging as a robust and accurate approach.

As highlighted earlier, the primary goal of this work is to seamlessly integrate this framework into an existing SOAR, underscoring the critical importance of detection rate and speed. Having successfully validated the detection rate, our future endeavors will focus on refining the Parallel UC2B framework by incorporating asynchronous communication capabilities. This enhancement aims to further reduce execution time while leveraging more nodes for superior strong and weak scalability. Additionally, we plan to conduct extensive testing across diverse applications such as healthcare and finance.

Acknowledgment

This research used computational resources of the supercomputer Fugaku provided by the RIKEN Center for Computational Science. We sincerely thank Research engineer Martial Mancip for his kind assistance, which greatly aided our study.

References

1. Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu Interconnect D. https://www.top500.org/system/179807/
2. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: 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. pp. 622–637. Springer (2019)
3. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special lecture on IE **2**(1), 1–18 (2015)
4. Anandakrishnan, A., Kumar, S., Statnikov, A., Faruquie, T., Xu, D.: Anomaly detection in finance: editors' introduction. In: KDD 2017 Workshop on Anomaly Detection in Finance. pp. 1–7. PMLR (2018)
5. Bukhari, O., Agarwal, P., Koundal, D., Zafar, S.: Anomaly detection using ensemble techniques for boosting the security of intrusion detection system. Procedia Computer Science **218**, 1003–1013 (01 2023). https://doi.org/10.1016/j.procs.2023.01.080
6. Cappello, F., Geist, A., Gropp, W., Kale, S., Kramer, B., Snir, M.: Toward exascale resilience: 2014 update. Supercomputing Frontiers and Innovations **1**(1), 5Äì28 (Jun 2014). https://doi.org/10.14529/jsfi140101, https://superfri.org/index.php/superfri/article/view/14
7. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. **41**(3) (jul 2009). https://doi.org/10.1145/1541880.1541882, https://doi.org/10.1145/1541880.1541882
8. Diop, A., Emad, N., Winter, T.: 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. pp. 101–110. IEEE (2020)
9. Diop, A., Emad, N., Winter, T.: 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. pp. 1–8. IEEE (2020)
10. Du, Q., Tang, B., Xie, W., Li, W.: Parallel and distributed computing for anomaly detection from hyperspectral remote sensing imagery. Proceedings of the IEEE **109**(8), 1306–1319 (2021). https://doi.org/10.1109/JPROC.2021.3076455
11. Emad, N., Petiton, S.G.: Unite and conquer approach for high scale numerical computing. J. Comput. Sci. **14**, 5–14 (2016)
12. Ghiasvand, S., Ciorba, F.M.: Anomaly detection in high performance computers: A vicinity perspective. In: 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC). pp. 112–120 (2019). https://doi.org/10.1109/ISPDC.2019.00024
13. Görnitz, N., Braun, M., Kloft, M.: Hidden markov anomaly detection. In: International conference on machine learning. pp. 1833–1842. PMLR (2015)
14. Humble, R., O'Shea, F.H., Colocho, W., Gibbs, M., Chaffee, H., Darve, E., Ratner, D.: Beam-based rf station fault identification at the slac linac coherent light source. Phys. Rev. Accel. Beams **25**, 122804 (Dec 2022). https://doi.org/10.1103/PhysRevAccelBeams.25.122804, https://link.aps.org/doi/10.1103/PhysRevAccelBeams.25.122804
15. Humble, R., Zhang, Z., O'Shea, F., Darve, E., Ratner, D.: Coincident learning for unsupervised anomaly detection (2023)

DOI: 10.1007/978-3-031-63783-4_26

16. Komolafe, T., Quevedo, A.V., Sengupta, S., Woodall, W.H.: Statistical evaluation of spectral methods for anomaly detection in static networks. Network Science **7**(2), 238–267 (2019)
17. Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al.: Long short term memory networks for anomaly detection in time series. In: Esann. vol. 2015, p. 89 (2015)
18. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set pp. 1–14 (01 2016). https://doi.org/10.1080/19393555.2015.1125974
19. Nakao, M., Ueno, K., Fujisawa, K., Kodama, Y., Sato, M.: Performance of the supercomputer fugaku for breadth-first search in graph500 benchmark. In: High Performance Computing: 36th International Conference, ISC High Performance 2021, Virtual Event, June 24 – July 2, 2021, Proceedings. p. 372–390. Springer-Verlag, Berlin, Heidelberg (2021)
20. Nehinbe, J.O.: A simple method for improving intrusion detections in corporate networks. In: Information Security and Digital Forensics: First International Conference, ISDF 2009, London, United Kingdom, September 7-9, 2009, Revised Selected Papers 1. pp. 111–122. Springer (2010)
21. Reed, D.A., Dongarra, J.: Exascale computing and big data. Communications of the ACM **58**(7), 56–68 (2015)
22. Sato, M., Ishikawa, Y., Tomita, H., Kodama, Y., Odajima, T., Tsuji, M., Yashiro, H., Aoki, M., Shida, N., Miyoshi, I., Hirai, K., Furuya, A., Asato, A., Morita, K., Shimizu, T.: Co-design for a64fx manycore processor and "fugaku". In: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 1–15 (2020). https://doi.org/10.1109/SC41405.2020.00051
23. Shanbhag, S., Wolf, T.: Accurate anomaly detection through parallelism. IEEE Network **23**(1), 22–28 (2009). https://doi.org/10.1109/MNET.2009.4804320
24. Stojanovic, L., Dinic, M., Stojanovic, N., Stojadinovic, A.: 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). pp. 1647–1652. IEEE (2016)
25. Syarif, I., Zaluska, E., Prugel-Bennett, A., Wills, G.: Application of bagging, boosting and stacking to intrusion detection. In: Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8. pp. 593–602. Springer (2012)
26. Ten, C.W., Hong, J., Liu, C.C.: Anomaly detection for cybersecurity of the substations. IEEE Transactions on Smart Grid **2**(4), 865–873 (2011). https://doi.org/10.1109/TSG.2011.2159406
27. TOP500: Top500 list - november 2020. https://www.top500.org/lists/top500/2020/11/ (2020), accessed: July 26, 2023
28. TOP500: Top500 list - june 2021. https://www.top500.org/lists/top500/2021/06/ (2021), accessed: July 26, 2023
29. Ukil, A., Bandyoapdhyay, S., Puri, C., Pal, A.: Iot healthcare analytics: The importance of anomaly detection. In: 2016 IEEE 30th international conference on advanced information networking and applications (AINA). pp. 994–997. IEEE (2016)
30. Zineb, Z., Nahid, E., Ahmed, B.: A novel approach to parallel anomaly detection: application in cybersecurity. In: 2023 IEEE International Conference on Big Data (BigData). pp. 3574–3583. IEEE (2023)