IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 May 2024, accepted 22 July 2024, date of publication 29 July 2024, date of current version 28 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3434629

RESEARCH ARTICLE

Advanced Machine Learning Based Malware Detection Systems

SONG-KYOO KIM[1], (Senior Member, IEEE), XIAOMEI FENG[1], HUSSAM AL HAMADI[2], (Senior Member, IEEE), ERNESTO DAMIANI[3], (Senior Member, IEEE), CHAN YEOB YEUN[3], (Senior Member, IEEE), AND SIVAPRASAD NANDYALA[4]

[1]Faculty of Applied Sciences, Macao Polytechnic University, Macau, SAR, China
[2]Engineering and Information Technology, University of Dubai, Dubai, United Arab Emirates
[3]Center for Cyber-Physical Systems, EECS Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[4]Secure Systems Research Center, Technology Innovation Institute, Abu Dhabi, United Arab Emirates

Corresponding author: Song-Kyoo Kim (amang@mpu.edu.mo)

This work was supported in part by Macao Polytechnic University (MPU) under Grant RP/FCA-04/2023.

ABSTRACT In the area of machine learning (ML) training data optimization through the construction of compact data, the focus of this paper is presented. The concept of compact data design, aimed at creating an optimized dataset that maximizes benefits without the need to manage a vast amount of complex data, is introduced. Improvements in the methods for optimizing ML training have been incorporated into the development of artificial intelligence (AI) systems. The introduction of understanding ML training datasets as a facet of Explainable AI (XAI), comprehensible to humans, has been made. Among the methods of XAI, the evaluation of input feature importance stands out as a way to enhance the accuracy of complex ML models. The innovative method of compact data design for optimizing ML training through dataset reduction is proposed. The performance of an ML-based malware detection system, along with its variant utilizing compact data, has been assessed, demonstrating the maintenance of 99% accuracy. By applying a 76% reduced input dataset, the speed of ML training with the novel compact data design could be maximized, suggesting that an ML system trained in this manner could achieve statistically equivalent accuracy with only 57% of the original data sample size.

INDEX TERMS Compact data, data reduction, machine learning, malware, security, data complexity, artificial intelligence, supervised learning, robust classification.

I. INTRODUCTION

Artificial Intelligence (AI) or machine learning (ML) have been utilized in various pattern recognition tasks, ranging from image processing to natural language processing. However, contemporary AI and ML models play a crucial role across a broad spectrum of applications [1], [2], [3]. Supervised learning which is a pivotal machine learning task involves learning a function that associates an input with an output, guided by example input-output pairs [4]. This process entails inferring a function from labeled training data, which comprises a collection of training examples [5]. Among ML training methodologies, supervised learning is the most widely adopted, finding application in the vast majority of AI systems. As a result, over the past few years, the demand for the adaption of additional techniques by these machine learning (ML) models has increased [6]. The Convolutional Neural Network (CNN) is recognized as one of the popular ML models for a variety of computer vision applications. Meanwhile, compact data design represents a conceptual approach aimed at designing an optimized dataset that delivers optimal benefits without the complexities associated with handling large volumes of data [7]. Hence, the compact data should contain the maximum knowledge

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed.

patterns at fine-grained level for effective and personalized utilization of bigdata systems [8], [9], [10]. Design of proper compact dataset is especially vital for developing the AI and the ML. Compact data learning is constructing the optimized training dataset which gives the statistically same ML accuracy but with the reduced data size. It has been emphasized that applications initially classified as malware ought to maintain their classification, given that solely features characteristic of benign applications are eliminated [11]. Detection systems for malware applications have been studied extensively, especially after the proliferation of smartphones [12], [13], [14], [15], [16]. The widespread adoption of Android operating system across numerous IoT devices has transformed Android-based IoT devices into significant targets for malware [17]. It has been recognized that traditional malware detection methods which depended on maintaining a database of malicious applications identified through the computation of application signatures exhibit limited efficacy in identifying new, previously unknown malware [16]. Consequently, a range of malware detection strategies, including the enhancement of ML algorithms, has been put forward [18], [19], [20], [21]. Research has been conducted on new static algorithms for malware detection on the Android platform [18], [19]. Permissions, component deployment, intent passing, API calls, and the heterogeneous information network (HIN) are utilized as significant features for characterizing Android applications [19], [20], [22]. The majority of ML-based techniques for detecting malware predominantly depend on features derived from both static and dynamic analysis of applications. While these ML-based approaches are effective, they have been identified as vulnerable to adversarial attacks [23]. Furthermore, some existing static detection methods employing artificial intelligence algorithms for malware focus exclusively on the Java code layer for extracting API features, neglecting the significant amount of malicious behavior that involves native layer code [17].

Explainable AI (XAI) is a machine learning technique in which the results of the solution could be understandable by humans [24]. It aims to create a suite of techniques that produce more explainable models whilst maintaining high performance levels [24], [25]. Evaluating the input feature importance is one of XAI methods to improve the accuracy of a complex ML model [25]. The XAI could be adapted to construct the compact data for ML training. Recent studies have propose various techniques for analyzing feature importance including Model Class Reliance [26], Shapely feature importance [27] and Leave-One-Covariate-Out [28]. Explainable AI techniques are capable to analyze the importance of input features during ML training and the values are calculated after completing the training phase [29]. Some of them require high level of accountability and thus transparency, for example, the medical sector. Explanations for machine decisions and predictions are thus needed to justify their reliability [24]. Although XAI has been widely studied for analyzing input features of ML training, The analysis of XAI is only completed after at least one ML training process. In the other hands, additional redundant training sessions are required for analyzing the input feature importance even before starting a ML training. Compact data learning (CDL) introduces a novel and applicable structure for enhancing a classification system by reducing the machine learning training data size [30]. It is aimed at enhancing model training efficiency which seeks to optimize runtime speed by diminishing the sample size and minimizing data features. Through the use of reduced sampling to decrease the dataset size and the implementation of a robust comparison and selection procedure for feature reduction methods, we have effectively tackled the difficulties associated with training models on expansive datasets. This approach entails the development of an optimized training dataset that maintains comparable Machine Learning accuracy while minimizing data volume [7]. Additionally, our methodology not only boosts runtime efficiency but also ensures negligible impact on the original accuracy performance. The outcomes of this study offer insightful understanding and pragmatic techniques for augmenting model training efficiency, particularly in situations involving data volume reduction and feature selection. This research provides innovative and unique ways to design compact data. CDL is not only included in down-sizing data but also in a whole process to handle compact ML training dataset. Unlike atypical XAI, this innovative method could dramatically reduce input data size for ML training even before executing ML training. The actual implementation of this innovative methodology has been adapted into the real ML training case. The ML training and evaluation for the malware detection show the efficiency of this innovative optimizing ML training.

The paper is organized as follows: Section II provides the mathematical background for constructing the compact data for machine learning training. This section deals with theoretical background for reducing both input features (Section II-B) and sample sizes (Section II-C). The actual application of this novel compact data design are provided on Section III. The ML training for the malware detection is considered as an adaptation of this innovative compact data design. In this section, The ML training setup and the reference ML algorithm for evaluation are described. Overall performance comparisons within optimizations with CDL feature and/or sample reductions are also explained in this section. Finally, the conclusion is included in Section IV.

II. PRELIMINARIES

Malign samples were initially extracted from 50 malware instances which were validated by VirusTotal and collected from the Androzoo database [31]. Only default factory applications in both idle and active states were used for the collection of benign data. Conversely, malign data were collected from 24 live malware instances from the Androzoo dataset, which were run in a Sandbox Environment. Each of the 50 malware instances was observed in action on

TABLE 1. Malware datasets.

Dataset / Classification	class 0 (Benign)	class 1 (Malign)
Training	18507	12904
Testing	4627	3226

an Android-based smartphone for the collection of malign samples. To mitigate biases associated with data originating from different sources, 60 trusted applications from the Google Play store and the system were selected as the source for both benign and malign samples [32]. This environment for generating and collecting both benign and malign samples has been extensively employed in diverse studies [32], [33]. In our conducted experiments, a total of 31,411 samples, comprising 1,942 input features, were collected, with the sample outputs being categorized into a binary classification (0-Benign, 1-Malign). Another set of samples was utilized as the training dataset to facilitate the comparison among various ML system variants. The testing dataset, comprising 7,953 samples, does not overlap with the training dataset, yet it shares the same 1,942 input features and the identical single binary output {0, 1} as the training dataset (see Table 1).

A. VARIOUS MACHINE LEARNING MODELS FOR MALWARE DETECTION

In this section, we provide an overview of our research methodology in this study. Various machine learning algorithms have been tested for selecting the best model for the credit card fraud detection. Among these models, we have chosen ensemble-based learning models and traditional machine learning models for our analysis [34], [35], [36], [37]. There are five ML algorithms which have applied into the same datasets. These ML models with the balanced datasets are considered for this research:

- **Random Forest (RF)** [34] constructs a stronger classifier through training Random Forest which is an ensemble learning model consist of multiple decision trees based on random feature selection and boot program [38]. This combined model adjusts the weights of samples based on the performance of the previous round's classifier and strengthens the training of misclassified samples in the next round. The pairing of AdaBoost with RF enhances its robustness and improves the quality of classification for imbalanced credit card data.
- **Gradient Boosted Decision Tree (GBDT)** [35] is an ensemble learning algorithm which iteratively training a series of decision trees to build a powerful predictive model. GBDT has also been used in previous papers as a base learner for fixed-size decision trees to overcome the problem of decision trees limiting their depth due to exponential growth.
- **K-Nearest Neighbor (KNN)** [36] is a model through voting its local neighbouring data points to build the classifier function [37], [39], [40]. User set the number of k and the 'neighbors' value is initially chosen randomly, but it can be fine-tuned through iterative evaluation.
- **Convolutional Neural Network (CNN)** [36] is a deep learning method that is widely used in images, text, audio and time series data, etc. There are six different layers in the CNN model including input layer, convolutional layer, pooling layer, fully connected layer, SoftMax/Logic layer and output layer, of which hidden layers with the same structure can have different numbers of channels per layer.
- **Support Vector Machine (SVM)** [37] utilizes both classification and regression tasks. SVM is known for its capability to derive optimal decision boundaries between classes. However, it is not well-suited for datasets exhibiting imbalanced class distribution, noise, and overlapping class samples.
- **Decision Tree (DT)** [41] use actual data attributes to create decision rules in a tree-like structure. They visually present information in an easily understandable tree pattern. The decision tree algorithm has the advantage of not requiring feature scaling, being resilient to outliers, and automatically handling missing values.

It is noted that the performance results for the above ML models are not the same as the results from the original research because all above researches are completed based on the unbalanced dataset.

B. VARIOUS FEATURE REDUCTION METHODS

Feature selection methods have gained widespread adoption in addressing high-dimensional problems due to their simplicity and efficiency [42]. Feature selection aids in data understanding, reduces computational demands, mitigates the curse of dimensionality, and enhances predictor performance [43]. The essence of feature selection lies in selecting a subset of variables from the input and effectively captures the input data while minimizing the influence of noise or irrelevant variables to generate robust predictive outcomes [43], [44].

- **Analysis of Variance (ANOVA)** is a statistical method used to compare means across different groups by analyzing data variance. It is commonly used in feature selection to aid in inference and decision-making processes. This method has reused in previous paper [45].
- **Feature importance method** is a technique used to evaluate and quantify the importance of features in a machine learning model, which helping user understanding the critical role of specific features in the predictive performance of a model.
- **Correlation heatmap** is a graphical representation that visualizes pairwise correlations between variables in a data set and is generated based on linear correlation coefficients. In the correlation heatmap, darker blue indicates a stronger negative correlation, while darker red indicates a stronger positive correlation.

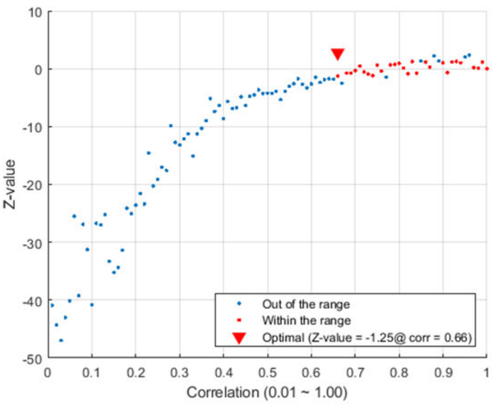

FIGURE 1. Experiments for the Z-values based on correlation changes.

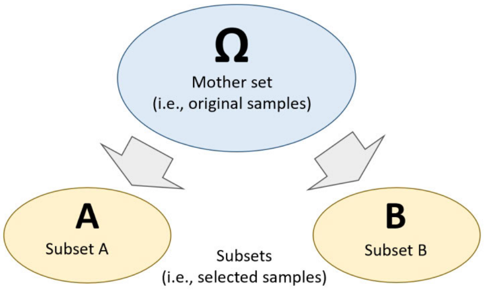

FIGURE 2. Set diagram for statistical sampling.

- **Linear correlation coefficient** is employed to quantify the strength and direction of the linear relationship between two variables [46].
- **Compact data learning (CDL)** (feature reduction only) is the enhanced feature reduction method which is based on the correlation, a correlation heapmap is directly applied to calculate the pair-wise comparison between the input features of the dataset [30]. CDL serves as a specific framework intended to accelerate the machine learning training phase without sacrificing system precision.

The above five feature reduction methods have been employed to select the features for training the machine learning models. The outputs of these methods are compared to determine the optimal feature reduction approach for further training. Resampling techniques are utilized to eliminate redundant data instances from the dataset. According to various trials of machine learning training, $r \geq 0.7$ under $\alpha = 0.1$ gives enough the resemblance which compare to an original input feature set. Hence, a default value of r^* shall be 0.7 (i.e., $r^* = 0.7$). Although this value is not the best solution but it provides faster training time than the original training because the number of input features is reduced.

C. SAMPLE REDUCTION BY USING CDL

The core concept revolves around determining the quantity of samples for which the probability distributions of subsets A and B mirror the probability distribution of the parent set Ω, despite the subsets' sample sizes being smaller than that of the mother set (see Fig. 2).

Let X be an input data which is the random variable based on the samples in a subset and the probability of a hypothesis test is defined as follows:

$$P\left\{\left|\frac{\overline{X}-\mu}{\frac{\sigma}{\sqrt{n}}}\right|<\epsilon\right\}>\beta,\ \Phi\left(\epsilon\right)=\beta, \tag{1}$$

and the cumulative distribution function (CDF) $\Phi(x)$ is the standard normal distribution as follows:

$$\Phi\left(z\right)=P\left\{Z\leq z\right\}=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z}e^{-\frac{z^2}{2}}dx, \tag{2}$$

where ϵ is a small number which is a tolerance of samples and β is the target probability for a resemblance between a mother sample mean and a sample mean. From (1) and (2), the minimum number of samples is as follows:

$$n_k \geq \left(\frac{z^*\cdot\sigma_k}{\overline{X}_k-\mu_k}\right)^2,\ z^*=\Phi^{-1}\left(\beta\right),$$

where $z^* \simeq 2.57$ when $\beta = 0.99$ and

$$\mu_k=\mathbb{E}\left[\overline{X}_k\right],\ \sigma_k^2=\mathbb{E}\left[\left(\overline{X}_k-\mu_k\right)^2\right], \tag{3}$$

which could be calculated from the mother set with n^0 samples. A typical value of β is 0.90 (i.e., 90% of resemblance), 0.95 (95% of resemblance) or 0.99 (99% of resemblance). Let $Q \in \left\{1, 2, \ldots, r^0\right\}$ be the set of the output classifications (i.e., types of an output). Each original sample (or trial) is maped with one of these classification within $\left\{1, 2, \ldots, r^0\right\}$ and r^0 is the number of classification outputs. Let Y_r be the (mother) set of the required sample sizes for the output classification l as follows from (4):

$$Y_r=\left\{n_1^l,n_2^l,\ldots,n_m^r\right\},\ l=1,\ldots l^0, \tag{4}$$

and

$$n_k^l=min\left(n_0^l,\left(\frac{z^*\cdot\sigma_k}{\overline{X}_k^l-\mu_k}\right)^2\cdot\mathbf{1}_{\left\{\overline{X}_k^l\neq\mu_k\right\}}\right), \tag{5}$$

$$\overline{X}_k^l=\left(\frac{1}{n_0^l}\right)\sum_{i=1}^{n^l}x_i,\ x_i\in X_k^r,\ n_0^r=n\left(X_k^l\right), \tag{6}$$

where X_k^l is the samples of the input feature k given the output classification l and n_0^l is the number of samples which are labeled with the output classification l:

$$n_l^*=\left\lceil\left(\frac{1}{m}\right)\sum_{k=1}^{m}n_k^l\right\rceil, \tag{7}$$

and $\delta(q) = \Phi^{-1}(q + 0.5)$. It is noted that δ is a tolerance rage which indicates how far from the mean in the standard normal distribution [30]. The function δ gives the z-value which indicates by adding probability q after passing the mean in the standard Normal distribution (i.e., $\Phi(0) = 0.5$).

TABLE 2. Performance results of various machine learning algorithms.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score	Training time (sec)
RF [34]	99.53	99.54	99.49	0.995	11.98
CNN [36]	99.10	99.15	98.91	0.991	2800.42
DT [41]	98.90	98.86	98.88	0.9887	4.59
KNN [36]	97.75	97.67	97.67	0.977	20.82
GBDT [35]	97.10	97.38	96.66	0.970	83.56
SVM [37]	95.17	96.13	94.17	0.949	691.98

TABLE 3. Accuracy comparisons of various feature reduction methods.

Algorithms	Original (%)	ANOVA (%)	Feat. Imp. (%)	Heatmap (%)
RF [34]	99.53	99.52	99.41	96.99
CNN [36]	99.10	98.96	97.67	95.28
DT [41]	98.90	98.62	98.74	96.40
KNN [36]	97.75	98.03	98.57	95.92
GBDT [35]	97.10	97.11	98.97	93.26
SVM [37]	95.17	95.10	94.59	89.79

III. EXPERIMENT RESULT

The upcoming discussion presents the optimization results of the five machine learning algorithms detailed in Section 2.5. Section III-A illustrates the performance metrics, including accuracy, precision, recall, f1-score, as well as the execution time of algorithms, when deployed on a balanced dataset using the testing dataset. Advancing to Section III-B, three feature reduction methods, specifically ANOVA, Feature Importance, and Correlation Heatmap, are implemented to generate accuracy results and determine the most effective feature reduction method. Conclusively, in Section III-C, we apply the feature reduction method chosen from Section III-B to conduct feature filtering during training, aiming to confirm the acceptability of the accuracy results. Simultaneously, we also record the execution time performance of the ML algorithms.

A. RESULT COMPARISONS FOR ML ALGORITHMS

The subsequent discussion features the performance results of the test dataset, which are also analyzed. Displayed in Table 9 are the original performance results for the five machine learning models. Notably, all models achieved performance scores ranging from 95.17% to 99.53% across the evaluation metrics. It is noted that the RF model achieved the highest performance across all accuracy related indicators but the random forest (RF) provides highest performance in terms of the training time. Among all, the CNN model exhibited the longest running time, requiring 47 minutes to complete.

B. RESULT COMPARISONS FOR FEATURE REDUCTION

Based on the accuracy results shown in Table 3, the ANOVA, feature importance and correlation heatmap methods have been utilized to reduce the number of features in the dataset, the training results of the models are presented in Table 3. In ANOVA, a significance level α of 0.05, representing a 95% confidence level has been selected. Subsequently, features with p-values lower than the established significance level have been selected for model training. In feature importance analysis, features with a score of zero have been removed as they have no predictive capability for the target variable.

Similarly, the features with a score of zero are not selected because they show no significant correlation with the target variable in the correlation heatmap. The accuracy scores of the correlation heatmap group have been consistently higher or equal to those of the original group. In contrast, while the ANOVA group and Feature Importance group have one or two accuracy scores that decreased. Consequently, the Correlation heatmap approach might be adapted for further model training. The accuracy results obtained using CDL for feature selection in machine model training are showcased in Table 4.

TABLE 4. Accuracy comparisons of CDL feature reduction method.

Algorithm	Original (%)	CDL ($r^* = 0.7, \alpha = 0.1$)		
		Accuracy (%)	H_0 Accepted?	Training Time (sec)
RF [34]	99.53	99.58	Yes	10.74
CNN [36]	99.10	98.97	Yes	2667.23
DT [41]	98.90	98.79	Yes	3.82
KNN [36]	97.75	97.68	Yes	15.32
GBDT [35]	97.10	96.78	Yes	63.53
SVM [37]	95.17	94.93	Yes	579.03

In this study, the null hypothesis assumes no significant accuracy difference between the two samples, at a significance level of 0.1. The findings indicate that when the correlation threshold r^* is 0.7, the GBDT model showed a slight accuracy improvement, while the other four models experienced a decrease in accuracy. By applying the Z-test method, it was found that only the KNN model had a Z-score exceeding the critical value, leading to the rejection of the null hypothesis, implying that the accuracy of the KNN model is not acceptable. By using the Z-test method, it was determined that there is no significant accuracy difference among all models, leading to the acceptance of the null hypothesis, suggesting that the accuracy of all models is acceptable. The running times for all models are shorter than their original running times, indicating a decrease in running time achieved after feature reduction. From Table 3, the RF model showed an improvement in accuracy, while the GBDT, SVM, CNN, and KNN models experienced a decrease in accuracy but still acceptable (i.e., no differences). In the other The performance of the KNN model is unacceptable, as indicated by the rejection of the null hypothesis. The training times of all models were less than their original times. According to our experiments, the CDL based feature reduction is theoretically based on the absolute correlation and easily transformed

TABLE 5. Hyper-parameters for the malware detection datasets.

Parameter	Setup value	Description
m	1,942	Total number of initial input features
n_0	31,411	Total number of initial training samples
r^0	2	Number of output classifications
ϱ^*	0.7	Correlation threshold
$\delta(q)$	0.025	Tolerance rage

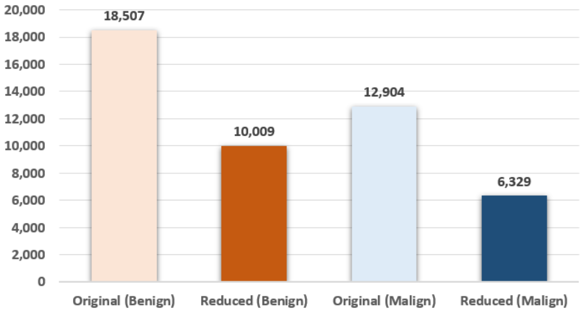

FIGURE 3. Optimized number of samples of the benign and malign training samples.

TABLE 6. Result comparison for various ML algorithms.

Algorithm	Accuracy (%)		Training Time (sec)
	Original	Sample Reduction	
RF [34]	99.53	99.25	5.29
CNN [36]	99.10	98.90	1368.07
DT [41]	98.90	98.42	2.30
KNN [36]	97.75	97.06	7.56
GBDT [35]	97.10	96.93	43.01
SVM [37]	95.17	93.63	205.58

from the correlation heatmap but this simple method is more practical and its efficiency is even better than the performance from a correlation heatmap. The figure presents the quantity and ratio of features prior to any feature removal, subsequent to feature selection using heatmap, and following feature selection using the CDL technique.

C. RESULT COMPARISONS FOR CDL SAMPLE REDUCTION

The theoretical methodology for reducing the sample size on the previous section is applied into the malware detection ML training. The samples are reduced to 16,338 from the original sample size (i.e., $n_0 = 31,411$ in Table 5).

The parameters for the CDL based sample reduction (e.g., $\delta(q)$, ϱ^*, r^0) in Table 5 are defined from the previous research [30]. The number of benign data size is 10,009 and 6,329 samples for malign data. It is noted that the number of the input features are not revised (i.e., $m = 1{,}942$ in Table 5). According to Fig. 4, the sample size has been reduced by 52%.

The CDL sample reduction is a simple but powerful method to optimize the training data size. The parameter for the sample reduction β could be arbitrary chosen as long as the Z-test has satisfy the hypothesis test. In our case, the proper value of the sample reduction parameter is 0.99 (i.e., $\beta = 0.99$).

TABLE 7. The CDL based feature reduction performance comparisons.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score
RF [34]	99.58	99.58	99.55	0.996
CNN [36]	98.97	99.05	98.82	0.989
DT [41]	98.79	98.75	98.75	0.9875
KNN [36]	97.68	97.58	97.63	0.976
GBDT [35]	96.78	97.10	96.26	0.966
SVM [37]	94.93	95.95	93.88	0.947

TABLE 8. Performance comparisons of different correlation score limits.

Algorithm	Original (%)	CDL ($r^* = 0.7, \alpha = 0.1$)		
		Accuracy (%)	H_0 Accepted?	Training Time (sec)
RF [34]	99.53	99.39	Yes	5.10
GBDT [35]	97.10	96.79	Yes	33.19
CNN [36]	99.10	98.60	No	1240.16
DT [41]	98.90	98.22	No	1.65
KNN [36]	97.75	96.96	No	5.11
SVM [37]	95.17	93.80	No	171.89

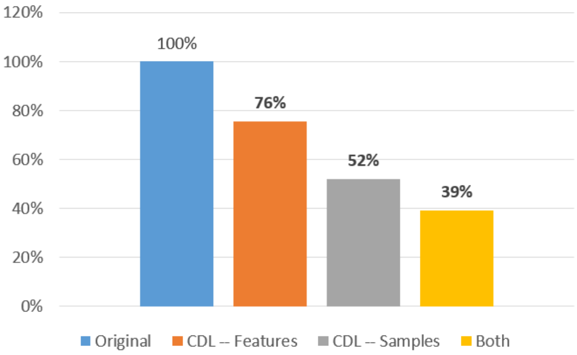

FIGURE 4. Performance comparison for the malware detection.

Besides of the accuracy measure, various performance measures for adapting the CDL based feature reduction have been evaluated (see Table 7). Our experiment indicates that CDL based feature reduction provides the better results of most performance measures including accuracy which compare to the ANOVA based feature reduction which is one of most widely feature reduction method (see Table 10 in Appendix A).

Since this experiment has been executed separately as a second round, each accuracy of the ML algorithms in Table 7 may not be the the same as the accuracy measures from the previous experiment in Table 6.

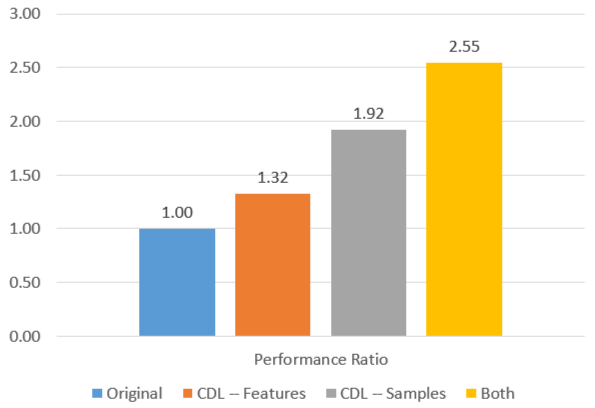

FIGURE 5. Performance comparison for the malware detection.

TABLE 9. Combined feature and sample reduction by using CDL.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score	Training time (sec)
RF [34]	99.39	99.43	99.31	0.994	5.10
GBDT [35]	96.79	97.14	96.28	0.967	33.19

IV. CONCLUSION

The CDL methods could optimize a ML training dataset without additional training sessions. Total of 34,111 malware samples with 1,942 input features are trained as the reference and various compact data have been constructed variously with the statistically same accuracy. It is found that the ML training dataset could be minimized by reducing the input features of 1,467 and the samples of 16,339, which corresponds to 2 times faster than the reference with maintaining 99% the accuracy. Although the CDL for combined feature and sample reduction provides the better performance in terms of accuracy and the training time for most of ML algorithms, this combined CDL reduction could not be applied on some ML models because of significant differences with the original ML algorithms.

According to our experiment result in Table 8, the random forest (RF) and the gradient boosted decision tree (GBDT) could be applied with the combined CDL (i.e., feature and sample reductions together). The ML algorithms shall give the statistically equal for various performance measures even after significant data reduction by adapting the combined CDL. Digesting the huge size of training dataset for ML systems have been a vital issue to breakthrough from traditional systems. However, current mechanisms including XAI require one or several additional ML training rounds before optimizing a ML algorithm. The novel compact data design which uses basic statistical methods including the correlation and the hypothesis test has been newly proposed. The data size of ML training is dramatically reduced based on the compact data learning (CDL) feature and/or sample reductions. The original data size is reduced by up to 76% for reducing the input features and 52% for reducing sample size. If both optimizations are applied, the training data size is reduced up to 39% (see Fig. 4).

TABLE 10. The ANOVA based feature reduction performance comparisons.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score
RF [34]	99.25	99.30	99.15	0.992
CNN [36]	98.90	98.97	98.77	0.989
DT [41]	98.42	98.41	98.32	0.9837
KNN [36]	97.06	97.07	96.85	0.966
GBDT [35]	96.93	97.26	96.44	0.968
SVM [37]	93.63	95.27	92.69	0.936

The compact data design for ML training provides the statistically same accuracy of the original malware detection by using only one of third of the original training data size (see Fig. 4). Roughly speaking, the ML training performance is 2 times faster than the original one after adapting CDL.

Based on our experiments, the CDL performs three times better for most ML algorithms in terms of the training times without scarify other major performance measures including accuracy, precision, recall and f1-score (see Table 9).

The efficacy of a ML-based malware detection system, including a version that employs compact data, was evaluated, revealing that it could sustain an accuracy of 99%. By reducing the input features by 76%, the training velocity of the ML system utilizing the innovative compact data approach was significantly enhanced. This indicates that an ML system trained using this method is capable of attaining an accuracy that is statistically equivalent to that achieved with only 57% of the original dataset size. Furthermore, the implementation of the combined CDL enables the attainment of comparable ML performance levels, despite a substantial data reduction of up to 60% although this efficiency is only applicable for selected ML models.

APPENDIX A
ANOVA BASED FEATURE REDUCTION PERFORMANCES

In this appendix, a comparison of the performance of various ML algorithms is provided by adapting ANOVA (Analysis of Variance) for feature reduction (see Table 10). ANOVA is widely employed in feature selection to support inference and decision-making processes [45]. This appendix covers the examination of diverse performance metrics for multiple ML algorithms following the implementation of feature reduction via ANOVA adaptation.

As mentioned in the previous section, it is observed that feature reduction by adapting CDL enhances the performance across all metrics, including accuracy, precision, recall, and f1-score (see Table 7 in Section III for comparison).

APPENDIX B
ALGORITHMS FOR CDL FEATURE REDUCTION AND SAMPLE REDUCTION

This appendix provides the the algorithm for the CDL based feature reduction which has been introduced from the

Algorithm 1 CDL Based Feature Reduction Algorithm [47]

```
Input: Correlation heatmap matrix CM, Score line SL
Output: Selected features SF
for scores in CM ($S_{ij} \in$ CM) do
    CM ← abs($S_{ij}$)
end
if CM ($S_{ij}$) ≥ SL then
    CM ← CM($S_{ij}$)
end
while length of CM ≠ 0 do
    $S_i$_most ← $S_i$ most frequency feature
    $S_j$_most ← $S_j$ most frequency feature
    if $S_i$_most ≥ $S_j$_most then
        CM ← CM delete all $S_i$_most
        print SF($S_i$_most)
    else
        CM ← CM delete all $S_j$_most
        print SF($S_j$_most)
    end
end
```

Algorithm 2 CDL Based Sample Reduction Alogorithm

```
Input: Training dataset D, Beta level $\beta$, Category number C
Output: Sample size $R_{sam}$, Optimized dataset $R_{data}$,
Effectiveness E, Maximum sample size $N_{max}$
n, m ← shape(D)
$D_{out}$ ← D(m)
$D_{in}$ ← D(0:m-1)

z ← $F^{-1}\left(\frac{1+\beta}{2}\right)$
m ← mean(D)
$\sigma$ ← standard_deviation($D_{in}$)
$I_{zero}$ ← index($\sigma$=0)
$D_{out\_u}$, $C_{out}$ ← $unique(D_{out})$

if $C = 0$ then
    $C \leftarrow length(D_{out})$
end

for $C_i$ in range(C) do
    $I \leftarrow index(D_{out} = D_{out\_u}[C_i])$
    $N_{star} \leftarrow (z * \sigma)^2/(mean(rows(D_{in}[I])) - m)^2$
    $N_{star}[I_{zero}] \leftarrow 0$
    $N_{max} \leftarrow length(I)$
    $I_{max} \leftarrow index(N_{star} > N_{max})$
    $N_{star}[I_{max}] \leftarrow N_{max}$
    $L \leftarrow N_{bar} - sd\sqrt{mean(N_{star} - N_{bar})^2}$
    $D_{out\_rev} \leftarrow rows(rows(D, I), 0 : int(L))$
end
$R_{sam} \leftarrow [L]$
$R_{data} \leftarrow array(D_{out\_u})$
E ← $\frac{sum([L])}{n}$
return $R_{sam}$, $R_{data}$, E, $N_{max}$
```

previous research (see Algorithm 1) [47]. Additionally, this section also provides the the algorithm for the CDL based sample reduction. Although basic theoremtical background of the CDL based sample reduction has been introduced from the previous research [30], the implementation algorithm for the sample reduction is never been introduced before (see Algorithm 2).

ACKNOWLEDGMENT

This article was revised using AI/ML-assisted tools. Special thanks to the reviewers who provide valuable advice for improving this paper.

REFERENCES

[1] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf., Comput. Commun. Secur.*, New York, NY, USA, 2006, pp. 16–25.

[2] Z. Xu and J. H. Saleh, "Machine learning for reliability engineering and safety applications: Review of current status and future opportunities," *Rel. Eng. Syst. Saf.*, vol. 211, Jul. 2021, Art. no. 107530.

[3] H. Olufowobi, R. Engel, N. Baracaldo, L. A. D. Bathen, S. Tata, and H. Ludwig, "Data provenance model for Internet of Things (IoT) systems," in *Service-Oriented Computing*. Berlin, Germany: Springer-Verlag, 2016, pp. 85–91.

[4] P. Norvig and S. J. Russell, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2010.

[5] A. Talwalkar, M. Mohri, and A. Rostamizadeh, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.

[6] M. A. Ramirez, S.-K. Kim, H. Al Hamadi, E. Damiani, Y.-J. Byon, T.-Y. Kim, C.-S. Cho, and C. Yeob Yeun, "Poisoning attacks and defenses on artificial intelligence: A survey," 2022, *arXiv:2202.10276*.

[7] S. A. Kim, "Toward compact data from big data," in *Proc. 15th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Dec. 2020, pp. 1–5.

[8] J. Dean, *Big Data, Data Mining, and Machine Learning*. Hoboken, NJ, USA: Wiley, 2014.

[9] K. Battams, "Stream processing for solar physics: Applications and implications for big solar data," 2014, *arXiv:1409.8166*.

[10] K. Kambatla, G. Kollias, V. Kumar, and A. Grama, "Trends in big data analytics," *J. Parallel Distrib. Comput.*, vol. 74, no. 7, pp. 2561–2573, 2014.

[11] S. Singh and G. Kaiser, "Metamorphic detection of repackaged malware," in *Proc. IEEE/ACM 6th Int. Workshop Metamorphic Test. (MET)*, Jun. 2021, pp. 9–16.

[12] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "DL-Droid: Deep learning based Android malware detection using real devices," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101663.

[13] X. Zhang, Y. Zhang, M. Zhong, D. Ding, Y. Cao, Y. Zhang, M. Zhang, and M. Yang, "Enhancing state-of-the-art classifiers with api semantics to detect evolved Android malware," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2020, pp. 757–770.

[14] S. Singh, B. Mishra, and S. Singh, "Detecting intelligent malware on dynamic Android analysis environments," in *Proc. 10th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Dec. 2015, pp. 414–419.

[15] S. Singh, S. Singh, and B. Mishra, "Artificial user emulator to detect intelligent malware on Android," *Int. J. Intell. Comput. Res.*, vol. 6, pp. 640–646, 2015.

[16] T. Alsmadi and N. Alqudah, "A survey on malware detection techniques," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Jul. 2021, pp. 371–376.

[17] R. Xixuan, Z. Lirui, W. Kai, X. Zhixing, H. Anran, and S. Qiao, "Android malware detection based on heterogeneous information network with cross-layer features," in *Proc. 19th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2022, pp. 1–4.

[18] O. Yildiz and I. A. Doğru, "Permission-based Android malware detection system using feature selection with genetic algorithm," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 29, no. 2, pp. 245–262, 2019.

[19] D.-J. Wu, C.-H. Mao, T.-E. Wei, H.-M. Lee, and K.-P. Wu, "DroidMat: Android malware detection through manifest and API calls tracing," in *Proc. 7th Asia Joint Conf. Inf. Secur.*, Aug. 2012, pp. 62–69.

[20] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu, "Make evasion harder: An intelligent Android malware detection system," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 5279–5283.

[21] W. C. Wu and S. H. Hung, "DroidDolphin: A dynamic Android malware detection framework using big data and machine learning," in *Proc. Conf. Res. Adapt. Convergent Syst. (RACS)*, New York, NY, USA, 2014, pp. 247–252.

[22] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," *J. Comput. Virol. Hacking Techn.*, vol. 12, pp. 59–67, Jun. 2016.

[23] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS)*, New York, NY, USA, 2017, pp. 506–519.

[24] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[25] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical XAI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, Nov. 2021.

[26] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a Variable's importance by studying an entire class of prediction models simultaneously," 2018, *arXiv:1801.01489*.

[27] G. Casalicchio, C. Molnar, and B. Bischl, "Visualizing the feature importance for black box models," in *Machine Learning and Knowledge Discovery in Databases*, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, Eds. Cham, Switzerland: Springer, 2019, pp. 655–670.

[28] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *J. Amer. Stat. Assoc.*, vol. 113, no. 523, pp. 1094–1111, Jul. 2018.

[29] A. Y. Al Hammadi, C. Y. Yeun, E. Damiani, P. D. Yoo, J. Hu, H. K. Yeun, and M.-S. Yim, "Explainable artificial intelligence to evaluate industrial internal security using EEG signals in IoT framework," *Ad Hoc Netw.*, vol. 123, Dec. 2021, Art. no. 102641.

[30] S. K. Kim, "Compact data learning for machine learning classifications," *Axioms*, vol. 13, no. 3, p. 137, Feb. 2024.

[31] K. Allix, T. F. Bissyandé, J. Klein, and Y. L. Traon, "AndroZoo: Collecting millions of Android apps for the research community," in *Proc. IEEE/ACM 13th Work. Conf. Mining Softw. Repositories (MSR)*, New York, NY, USA, May 2016, pp. 468–471.

[32] F. Wang, H. Al Hamadi, and E. Damiani, "A visualized malware detection framework with CNN and conditional GAN," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2022, pp. 6540–6546.

[33] F. Maasmi, M. Morcos, H. Al Hamadi, and E. Damiani, "Identifying applications' state via system calls activity: A pipeline approach," in *Proc. 28th IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Nov. 2021, pp. 1–6.

[34] E. Ileberi, Y. Sun, and Z. Wang, "Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021.

[35] T. M. Alam, K. Shaukat, I. A. Hameed, S. Luo, M. U. Sarwar, S. Shabbir, J. Li, and M. Khushi, "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201173–201198, 2020.

[36] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022.

[37] S. N. Kalid, K.-H. Ng, G.-K. Tong, and K.-C. Khor, "A multiple classifiers system for anomaly detection in credit card data with unbalanced and overlapped classes," *IEEE Access*, vol. 8, pp. 28210–28221, 2020.

[38] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, pp. 14277–14284, 2018.

[39] L. Rokach and O. Maimon, *Data Mining Knowl. Discovery Handbook*. New York, NY, USA: Springer, 2010.

[40] R. Taghizadeh-Mehrjardi, K. Nabiollahi, B. Minasny, and J. Triantafilis, "Comparing data mining classifiers to predict spatial distribution of USDA-family soil groups in Baneh region, Iran," *Geoderma*, vols. 253–254, pp. 67–77, Sep. 2015.

[41] J. K. Afriyie, K. Tawiah, W. A. Pels, S. Addai-Henne, H. A. Dwamena, E. O. Owiredu, S. A. Ayeh, and J. Eshun, "A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions," *Decis. Anal. J.*, vol. 6, Mar. 2023, Art. no. 100163.

[42] S. Akogul, "A novel approach to increase the efficiency of filter-based feature selection methods in high-dimensional datasets with strong correlation structure," *IEEE Access*, vol. 11, pp. 115025–115032, 2023.

[43] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[44] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Comput. Biol. Chem.*, vol. 32, no. 1, pp. 29–38, 2008.

[45] C. Wu, Y. Yan, Q. Cao, F. Fei, D. Yang, X. Lu, B. Xu, H. Zeng, and A. Song, "SEMG measurement position and feature optimization strategy for gesture recognition based on ANOVA and neural networks," *IEEE Access*, vol. 8, pp. 56290–56299, 2020.

[46] J. Biesiada and W. Duch, "Feature selection for high-dimensional data—A Pearson redundancy based filter," in *Computer Recognition Systems*. Berlin, Germany: Springer, 2007, pp. 242–249.

[47] X. Feng and S.-K. Kim, "Novel machine learning based credit card fraud detection systems," *Mathematics*, vol. 12, no. 12, p. 1869, Jun. 2024.

SONG-KYOO (AMANG) KIM (Senior Member, IEEE) received the M.S. degree in computer engineering and the Ph.D. degree in operations research from Florida Institute of Technology, in 1999 and 2002, respectively. He is currently an Associate Professor of computing program with Macao Polytechnic University. He used to be an Associate Professor of several United Arab Emirates universities. Before moving to Gulf Region, he was a Core Faculty Member with Asian Institute of Management, providing courses in technology, innovation, and operations. Before his academic career, he was the Technical Manager of the Mobile Communications Division, Samsung Electronics, for more than ten years and mainly dealt with technology management in the information technology industry. He has been an Invited Speaker at many international conferences concerning technology management, innovation process, operations research, and data sciences. He is also an External Reviewer of IEEE Transactions on Network Science and Engineering.

XIAOMEI FENG received the B.S. degree in software engineering from the University of Macau, in 2014, and the M.S. degree in business administration from Chaminade University, in 2020. She is currently pursuing the Ph.D. degree in computer applied technology with Macao Polytechnic University. She has prior experience working in the banking industry. Her research interests include machine learning, business process management, data modeling, and artificial intelligence.

HUSSAM AL HAMADI (Senior Member, IEEE) received the degree in computer engineering from Ajman University, in 2005, and the Ph.D. degree in computer engineering from Khalifa University, in 2017. From 2005 to 2010, he was a Computer Consultant and a Tutor in several governmental and private institutions and eventually joined Khalifa University, as a Teaching Assistant, in 2010. He is currently an Assistant Professor of engineering information technology with the University of Dubai. He holds several international certificates in networking, business, and tutoring, such as MCSA, MCSE, CCNA, CBP, and CTP. He is also a Research Scientist with the Center for Cyber-Physical Systems (C2PS), Khalifa University. His research interests include applied security protocols for several systems, such as software agents, SCADA, e-health systems, and autonomous vehicles.

ERNESTO DAMIANI (Senior Member, IEEE) is currently a Full Professor with the Università degli Studi di Milano, Italy, the Senior Director of the Robotics and Intelligent Systems Institute, and the Director of the Center for Cyber-Physical Systems (C2PS), Khalifa University, United Arab Emirates. He is also the Leader of the Big Data Area, Etisalat British Telecom Innovation Center (EBTIC), and the President of the Consortium of Italian Computer Science Universities (CINI). He is also a part of the ENISA Ad-Hoc Working Group on Artificial Intelligence Cybersecurity. He has pioneered model-driven data analytics. He has authored more than 650 Scopus-indexed publications and several patents. His research interests include cyber-physical systems, big data analytics, edge/cloud security and performance, artificial intelligence, and machine learning. He was a recipient of the Research and Innovation Award from the IEEE Technical Committee on Homeland Security, the Stephen Yau Award from the Service Society, the Outstanding Contributions Award from IFIP TC2, the Chester-Sall Award from IEEE IES, the IEEE TCHS Research and Innovation Award, and the Doctorate Honoris Causa from INSA-Lyon, France, for his contribution to big data teaching and research.

CHAN YEOB YEUN (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in information security from the Royal Holloway, University of London, in 1996 and 2000, respectively. After the Ph.D. degree, he joined Toshiba TRL, Bristol, U.K., and later, he became the Vice President of the Mobile Handset Research and Development Center, LG Electronics, Seoul, South Korea, in 2005. He was responsible for developing mobile TV technologies and related security. He left LG Electronics, in 2007, and joined ICU (merged with KAIST), South Korea, until August 2008, and then the Khalifa University of Science and Technology, in September 2008. He is currently a Researcher of cybersecurity, including the IoT/USN security, cyber-physical system security, cloud/fog security, and cryptographic techniques; an Associate Professor with the Department of Electrical Engineering and Computer Science; and the Cybersecurity Leader of the Center for Cyber-Physical Systems (C2PS). He also enjoys lecturing for the M.Sc. degree in cyber security and the Ph.D. degree in engineering courses with Khalifa University. He has published more than 140 journal articles and conference papers, nine book chapters, and ten international patent applications. He also works on the editorial board of multiple international journals and on the steering committee of international conferences.

SIVAPRASAD NANDYALA received the Ph.D. degree in speech processing. He is currently a Senior Machine Learning Engineer with the Technology Innovation Institute. He is an innovator at heart, his work encompasses developing cutting-edge solutions for data security, anomaly detection, and energy management, backed by a rich portfolio of patents and publications. He has experience across multiple high-tech industries, he stands at the forefront of leveraging AI to address contemporary challenges.