ASTES

Advances in Science, Technology & Engineering Systems Journal

VOLUME 9-ISSUE 6|NOV-DEC 2024

www.astesj.com
ISSN: 2415-6698

Editorial

The evolving intersections of artificial intelligence, human-centered systems, industrial optimization, and security underscore the critical importance of interdisciplinary research in addressing pressing global challenges. This issue brings together innovative work spanning machine learning applications in manufacturing, vision-based health monitoring for elder care, educational tools for lean management, and robustness in deep learning security—each offering impactful insights into how intelligent systems are transforming industries and lives.

Manufacturing continues to benefit from the integration of advanced machine learning, particularly in custom industrial production environments. A regression-based approach using gradient-boosting techniques—specifically LightGBM and XGBoost—demonstrates notable performance improvements in predicting work man-hours for metal sheet stamping projects. These models are enhanced through strategic feature engineering, selection, and synthetic data generation. Results highlight not only the high accuracy of the optimized models but also the importance of interpretability through explainable AI tools, which empower domain experts to critique and trust machine-generated predictions [1].

The application of AI in healthcare is exemplified through a deep learning-based fall detection system designed to support elderly individuals, a demographic increasingly in need of non-invasive monitoring solutions. By leveraging Human Silhouette and Dense Optical Flow techniques, the system creates a robust visual representation of motion, further refined by convolutional and recurrent neural networks. This multi-modal approach significantly improves detection accuracy, achieving a remarkable 99% on a benchmark dataset. The solution provides a scalable, vision-based tool for fall detection that accounts for complex human movements and varied camera perspectives, offering substantial promise for enhancing elderly care [2].

A more educational and conceptual contribution focuses on improving understanding of “value” in lean management through a novel gamified approach. Value Karuta (VK), inspired by the traditional Japanese card game, acts as an engaging educational medium for communicating foundational lean principles. Survey-based evaluations in both Japan and the UK confirm the tool’s efficacy in teaching the abstract concept of customer value. Combining qualitative observation with quantitative feedback, the study validates VK’s role in strengthening conceptual understanding among students, professionals, and academics, contributing meaningfully to the pedagogy of lean thinking [3].

Security and reliability in artificial intelligence are critical, particularly as deep learning models become more embedded in real-world systems. A focused investigation into quantized neural networks explores their resilience to adversarial attacks, a significant vulnerability in conventional deep models. Quantization is shown to provide inherent robustness, reducing the efficacy of perturbation-based attacks even in scenarios where adversarial examples typically succeed. To support ongoing work in this domain, the authors release the Adversarial Neural Network Toolkit (ANNT), encouraging reproducibility and further exploration of security techniques in neural network compression [4].

Together, these papers illuminate how modern computational techniques can be leveraged for practical solutions across diverse domains. From optimizing industrial workflows to safeguarding AI against manipulation, and from enhancing elder care to reimagining business education, these studies reflect a shared commitment to purposeful and resilient technological development.

References:

[1] A. Emin U¨ nal, H. Boyar, B. Kuleli Pak, V.C. C¸ ag˘rı Gu¨ngo¨r, “Utilizing 3D models for the Prediction of Work Man-Hour in Complex Industrial Products using Machine Learning,” Advances in Science, Technology and Engineering Systems Journal, 9(6), 1–11, 2024, doi:10.25046/aj090601.

[2] W. Pa Pa San, M. Khaing, “Advanced Fall Analysis for Elderly Monitoring Using Feature Fusion and CNN-LSTM: A Multi-Camera Approach,” Advances in Science, Technology and Engineering Systems Journal, 9(6), 12–20, 2024, doi:10.25046/aj090602.

[3] T. Kobayashi, K. Murata, “Development and Application of Value Karuta to Understand Value in Lean Management: Initial Small-group Trial in Japan and the UK,” Advances in Science, Technology and Engineering Systems Journal, 9(6), 21–29, 2024, doi:10.25046/aj090603.

[4] A. Shrestha, J. Großmann, “On Adversarial Robustness of Quantized Neural Networks Against Direct Attacks,” Advances in Science, Technology and Engineering Systems Journal, 9(6), 30–46, 2024, doi:10.25046/aj090604.

Editor-in-chief

Prof. Passerini Kazmersk

ADVANCES IN SCIENCE, TECHNOLOGY AND ENGINEERING SYSTEMS JOURNAL

Volume 9 Issue 6 November-December 2024

CONTENTS

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 9, No. 6, 01-11 (2024)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

Utilizing 3D models for the Prediction of Work Man-Hour in Complex Industrial Products using Machine Learning

Ahmet Emin Ünal[*,1,2], Halit Boyar[1], Burcu Kuleli Pak[1], Vehbi Çağrı Güngör[3]

[1]*R&D Department, Adesso Turkey, Istanbul, 34398, Turkey*

[2]*Dept. of Comp. Engineering, Istanbul Technical University, Istanbul, 34485, Turkiye*

[3]*Dept. of Comp. Engineering, Abdullah Gül University, Kayseri, 38080, Turkiye*

ARTICLE INFO

Article history:
Received: 20 September, 2024
Accepted: 04 November, 2024
Revised: 05 November 2024
Online: 18 November, 2024

Keywords:
Complex Industrial Products
Metal Sheet Stamping
Work Man-hour Prediction
Machine Learning
Gradient Boosting

ABSTRACT

The integration of machine learning techniques in industrial production has the potential to revolutionize traditional manufacturing processes. In this study, we examine the efficacy of gradient-boosting machine learning models, specifically focusing on feature engineering techniques, applied to a novel dataset with 3D product models pertaining to work man-hours in metal sheet stamping projects, framed as a regression task. The results indicate that LightGBM and XGBoost surpass other models, and their effectiveness is further enhanced by employing feature selection and synthetic data generation methods. The optimized LightGBM model exhibited superior performance, achieving a MAPE score of 10.78%*, which highlights the effectiveness of gradient boosting mechanisms in handling heterogeneous data sets typical in custom manufacturing. Additionally, we introduce a methodology that enables domain experts to observe and critique the results through explainable AI visualizations.*

1. Introduction

This manuscript serves as an extension of a previous study on predicting work man-hours of complex industrial products, originally presented in 2023 4th International Informatics and Software Engineering Conference (IISEC) [1].

This study aims to contribute to the application of machine learning in industrial production by focusing on enhancing efficiency, productivity, and decision-making, specifically targeting work man-hour prediction in metal sheet stamping. By addressing this challenge, our research provides insights that fit within the broader scope of machine learning advancements in manufacturing. The integration of machine learning techniques in industrial production has the potential to revolutionize traditional manufacturing processes. Predictive systems for forecasting production and operational costs are crucial in shaping the future of machine learning applications in industrial production, and this study directly contributes to this important research area by focusing on work man-hour prediction.

In the field of complex industrial product management, where a custom configuration is needed for every product, accurately predicting the work man-hour for a product is essential for ensuring successful project completion. Rapid and precise responses to customer inquiries are crucial to maintaining competitiveness in the industry. However, given the complex and configurable nature of products, traditional methods of cost estimation may not provide the needed speed and accuracy. In the conventional approach, according to the domain knowledge of experts who shared the required dataset, they estimate the man-hour using customer requirements, 3D models, past similar projects, and a comprehensive analysis of the product. Traditional cost estimation methods have struggled to keep pace with the increased complexity and competitive environment of the industry, highlighting the need for more advanced approaches.

Recently, machine learning techniques have shown promising empirical results in improving the accuracy of various cost prediction models across many industrial sectors. This study builds upon these advancements by applying machine learning specifically to work man-hour prediction in the metal sheet stamping industry, addressing unique challenges in custom, short-run production. Recent studies have explored the application of machine learning in enhancing cost estimation in manufacturing processes. In [2], the authors applied back-propagation neural networks (BPN) and least squares support vector machines (LS-SVM) to address product life cycle cost estimation challenges, demonstrating the potential of ma-

[*]Corresponding Author: Ahmet Emin Ünal, Fax: +90 212 346 20 03, Phone: +90 212 346 20 02 & ahmet.unal@adesso.com.tr

chine learning in this area. Similarly, in [3], authors emphasized the importance of selecting a standard set of attributes for developing machine learning models for building project cost estimation, showcasing the advancements that machine learning offers in accurate cost estimation within the construction sector.

Research in [4] focused on explainable artificial intelligence for manufacturing cost estimation and machining feature visualization, indicating a growing interest in deep learning approaches for estimating manufacturing costs. In [5], authors proposed the use of two-dimensional (2D) and three-dimensional (3D) convolutional neural networks (CNN) for manufacturing cost estimation, highlighting the potential of deep learning methods in this context. In [6], authors explored early cost estimation in customized furniture manufacturing using machine learning, showcasing the application of machine learning for estimating costs in specific manufacturing niches. Furthermore, [7] discussed how intelligent job shop scheduling (JSS) systems, powered by machine learning and artificial intelligence solutions, aim to reduce costs based on specific cost functions, such as making span or economic cost. Additionally, [8] conducted an empirical study in the automotive industry, where they proposed machine learning as an advanced cost estimation method.

The use of neural networks in machining operations has been highlighted as advantageous in reducing uncertainties within the cost estimation process. In [9], authors emphasized the capacity of neural networks to enhance cost estimation accuracy in machining operations, showcasing the potential of machine learning in refining cost estimation models. In [10], authors compared various machine learning methods for estimating the manufacturing cost of jet engine components, displaying the effectiveness of different machine learning approaches in cost estimation for the aerospace industry. Moreover, in [11], authors developed methods for direct cost estimation in manufacturing parts, with recent studies leveraging deep learning techniques to predict manufacturing costs based on 3D CAD models. Additionally, [12] highlighted how machine learning improves prediction performance in surface generation and roughness in ultraprecision machining, emphasizing the role of machine learning in advancing automation and digitization in manufacturing processes.

Forecasting the work man-hour for producing complex industrial products poses distinct challenges. In contrast to conventional manufacturing methods that entail bulk production of identical units, often running into thousands or millions, metal sheet stamping operations are frequently tailored with short-run, tailored orders. This variability in design, materials, and processes complicates work man-hour estimations. Furthermore, the time-sensitive nature of such projects, combined with intense industry competition, demands swift and accurate cost estimates. The automotive sector serves as an example, predominantly employing manufacturing through sheet metal stamping projects [13].

The reliance on custom orders in the metal sheet stamping industry results in significant variability between projects, often leading to discrepancies in cost estimations. This variability complicates accurate cost prediction and underscores the importance of developing advanced estimation methods to mitigate financial and operational risks. An inaccurate prediction not only affects the financial bottom line but can also disrupt the broader supply chain, delay projects, and damage client relationships. In the worst cases, it may cause the rejection of profitable projects due to overestimated costs or the acceptance of unprofitable ones due to underestimations.

In the context of sheet metal stamping, where high production rates and cost-effectiveness are crucial factors, inaccurate cost predictions can result in suboptimal decision-making regarding material selection, tooling design, and process optimization [14]. This can lead to increased scrap rates, rework, and overall inefficiencies in the production line. Moreover, inaccurate cost predictions may also affect the competitiveness of manufacturers in the market, as cost overruns can erode profit margins and hinder the ability to offer competitive pricing [15]. Furthermore, inaccurate cost predictions in metal sheet stamping can impact the overall sustainability of manufacturing operations. For instance, if the estimated costs do not align with the actual expenses incurred during the stamping process, it can lead to increased waste generation, energy consumption, and environmental impacts [16]. This can undermine efforts to improve the environmental performance of manufacturing processes and reduce the overall carbon footprint of sheet metal stamping operations. Moreover, inaccurate cost predictions can also affect the quality and reliability of stamped metal parts. Suboptimal cost estimations may result in compromises in material selection, tooling quality, or process parameters, leading to variations in part dimensions, surface finish, or mechanical properties [17]. This can ultimately impact the functionality and performance of the stamped components, leading to potential quality issues and customer dissatisfaction.

Accurate cost predictions are essential for ensuring the economic viability, operational efficiency, and sustainability of metal sheet stamping processes. Inaccuracies in cost estimations can lead to significant issues, such as poor cost control, reduced competitiveness, increased environmental impact, and compromised product quality. Accurate predictions are crucial to prevent these issues, ensuring manufacturers can make informed decisions, maintain market competitiveness, and promote sustainable practices. This study aims to address these challenges by leveraging advanced machine learning techniques to enhance cost estimation accuracy in the metal sheet stamping process. Therefore, leveraging advanced cost estimation methods, such as machine learning algorithms or finite element modeling, can help mitigate the risks associated with inaccurate cost predictions and optimize the overall performance of sheet metal stamping processes.

The integration of machine learning in cost estimation processes within the manufacturing sector has shown significant promise in enhancing accuracy, efficiency, and decision-making. From product life cycle cost estimation to customized furniture manufacturing and jet engine component cost estimation, machine learning methods have demonstrated their versatility and effectiveness in optimizing cost estimation models. As manufacturing industries continue to deploy advanced technologies, the role of machine learning in cost estimation will become even more pivotal in driving operational excellence and cost-effectiveness.

In this study, we examine the efficacy of gradient-boosting machine learning models, specifically focusing on feature engineering techniques. We apply these methods to a novel dataset related to work man-hours in metal sheet stamping projects, framing the problem as a regression task. The results indicate that LightGBM and XGBoost surpass other models, and their effectiveness is further improved by employing feature selection and synthetic data generation

techniques.

Our study utilizes gradient boosting machine learning models, known for their efficacy with tabular data, and uniquely incorporates domain-specific knowledge tailored to the metal sheet stamping industry. This integration of expert insights and historical data aims to capture the unique challenges of custom, short-run production, setting our approach apart from general-purpose cost estimation models. This approach aims to enhance the predictive accuracy by integrating insights from historical data and expert analysis, tailored specifically to the nuances of metal sheet stamping.

A significant advancement in gradient boosting is the development of the XGBoost algorithm, known for its scalability and efficiency in building tree boosting models [18]. XGBoost, an integrated learning technique utilizing the gradient boosting algorithm, has been successfully applied in diverse industrial domains. For example, it has been used in predicting power demand for industrial customers [19] and transforming the used car market by accurately predicting prices [20]. The robustness and performance of XGBoost have been demonstrated in various applications, underscoring its effectiveness in industrial cost prediction tasks. Furthermore, the application of gradient boosting in industrial contexts extends to addressing specific challenges in cost prediction and optimization. NGBoost, a gradient boosting approach utilizing Natural Gradient, has been developed to tackle technical challenges in probabilistic prediction, thereby enhancing the accuracy and reliability of predictive models [21]. Additionally, diversified gradient boosting ensembles have been employed for predicting the cost of forwarding contracts, showcasing the versatility of gradient boosting methods in effectively handling regression and classification problems [22].

In the realm of energy consumption modeling, gradient boosting machines have been utilized to model the energy consumption of commercial buildings, demonstrating improved prediction accuracy compared to traditional regression models and random forest algorithms [23]. Similarly, in the context of cargo insurance frequency prediction, XGBoost has shown superior accuracy compared to other machine learning models, highlighting the efficacy of gradient boosting in diverse industrial prediction tasks [24].

The utilization of gradient boosting algorithms, such as XGBoost and LightGBM, has significantly impacted industrial cost prediction by enhancing prediction accuracy, scalability, and robustness in diverse industrial settings. From energy consumption modeling to customer attrition prediction, gradient boosting has emerged as a powerful tool for optimizing predictive models and improving decision-making processes in industrial cost estimation and optimization tasks. This study examines the effectiveness of gradient boosting machine learning models as well as feature engineering strategies on a new dataset concerning work man-hours in a metal sheet stamping project, framed as a regression task. The results indicate that LightGBM and XGBoost yield better performance compared to other models, and that feature selection along with synthetic data generation enhance the outcomes. The main aims of this research are as follows:

1. Compare the performance of different machine learning models and feature engineering techniques for work man-hour prediction in metal sheet stamping projects.
2. Identify key variables and features that contribute to the accuracy of work man-hour predictions.
3. Assess the integration of industry-specific knowledge into machine learning models, evaluating its impact on predictive accuracy.

The structure of this paper is outlined as follows: Section II reviews the existing research on various methodologies for forecasting work man-hours in industrial projects; Section III provides an explanation of the utilized dataset; Section IV provides a detailed account of the model experiments conducted during the study; Section V presents a discussion of the experimental findings; and Section VI offers concluding remarks.

2. Related Works

Various studies across industrial fields such as automotive, construction, and furniture manufacturing have explored the prediction of production costs, labor costs, and material costs using diverse machine learning methods. The application of these techniques varies significantly based on the industry's specifics and the nature of the available data, highlighting the need for industry-specific adaptations of general methodologies.

In [25], authors employed several machine learning models on wheel cost data of 1340 automobiles. After implementing feature selection techniques, their findings revealed that Support Vector Regression (SVR) achieved the highest R^2 value in the cross-validation set. Interestingly, Linear Regression (LR) scored better in the test set, which may suggest that simpler models can sometimes outperform more complex ones in less volatile environments. This finding is relevant to our research as it underscores the importance of evaluating model complexity in the context of specific data characteristics, which is crucial for optimizing cost prediction accuracy in our own study.

Voxelization is a fundamental process in feature extraction for cost prediction tasks, especially in industrial production settings. It involves converting 3D CAD models or point cloud data into a structured voxel grid, which is particularly important for enabling deep learning models like Convolutional Neural Networks (CNNs) to effectively process and analyze complex geometries. This process is crucial in our research as it allows us to capture detailed geometric features that directly impact cost prediction accuracy, especially in scenarios involving intricate part designs. By using voxelization, we can ensure that our models effectively learn from the geometric complexity of the industrial components, leading to more precise predictions. Various studies in computer science and point cloud processing emphasize the importance of voxelization in processing point cloud data for tasks like object detection and feature extraction [26, 27]. In the field of mechanical parts manufacturing, authors of [28] innovatively applied Convolutional Neural Networks (CNN) to predict manufacturing costs. By utilizing voxelization to transform 3D models into a trainable format, they achieved a mean absolute percentage error ($MAPE$) of 6.34%. This approach underscores the potential of advanced image processing techniques in enhancing feature extraction for cost prediction models. Voxel-based methods have shown particular success in the aerospace industry as well, where converting complex 3D geometries of jet engine components into voxel grids allows for more accurate cost estimation and defect

detection. Techniques like Fully Convolutional Networks (FCN) and autoencoders, as discussed in [29] and [30], further enhance voxel-based feature extraction for tasks such as object detection and image processing.

The furniture manufacturing industry also demonstrates the importance of early cost estimation due to its highly customizable nature. In [31], authors compared various algorithms, such as Extra Trees Regressors (ETR), Gradient Boosting Regressors (GBR), and Random Forest (RF) on data from 1026 products of a Lithuanian furniture manufacturer. The RF algorithm exhibited superior performance, achieving an R^2 score of 0.84, which highlights the effectiveness of ensemble methods in handling heterogeneous data sets typical in custom manufacturing.

Random Forest, as a versatile machine learning algorithm, excels at handling high-dimensional data and capturing complex relationships, making it ideal for cost prediction and optimization in industrial production. Its robust performance in classification and regression tasks supports accurate component classification and production cost prediction, essential in custom manufacturing [32]. Additionally, Random Forest has been instrumental in developing efficient predictive maintenance systems, enabling organizations to anticipate equipment failures, optimize maintenance schedules, and improve production performance [33]. Its interpretability is particularly beneficial in environments where stakeholders must understand the factors influencing costs or production outcomes, aiding decision-making processes [34]. Furthermore, Random Forest's capability to manage complex interactions and highly correlated variables makes it well-suited for settings with intricate production processes and variable interdependencies [35]. Given its flexibility and strong adaptability in real-world applications, Random Forest is a reliable choice to improve production efficiency and optimize cost prediction in the landscape of custom manufacturing [36]. These studies supports our methodology by demonstrating the value of using ensemble methods like Random Forest to effectively manage variability and complexity, similar to the challenges faced in our study of cost prediction for metal sheet stamping.

Parallel to our focus, in [37], authors developed an early cost estimation model specifically for stamping dies, employing Artificial Neural Networks (ANN), which demonstrated a deviation of 8.28% on test data. This study exemplifies the applicability of ANN in industries where data can be nonlinear and complex. ANNs excel in nonlinear cost estimation for custom manufacturing due to their ability to capture complex nonlinear relationships between variables and costs which is vital to model intricate scenarios [38, 39]. They surpass traditional methods in prediction accuracy, thus optimizing schedules and enhancing decision-making processes [39, 40]. Furthermore, ANNs adapt flexibly to changing data patterns, effectively managing the intricacies of custom manufacturing cost data [41, 42]. They are adept at extracting relevant features from complex datasets and recognizing hidden patterns, which is crucial for optimizing cost estimation models [43, 44]. Despite their complexity, efforts to enhance the interpretability of ANNs help provide transparency in the decision-making process [45, 44]. Ensemble methods involving ANN improve prediction accuracy and reduce errors, thus bolstering the robustness and reliability of models [46]. Furthermore, ANNs demonstrate excellent generalization to unseen data and maintain robust performance in diverse scenarios [47], significantly enhancing cost estimation processes, optimizing resource allocation, and supporting decision making in custom manufacturing environments. The capabilities of ANNs justify their use as a benchmark in managing the complex interactions and nonlinear relationships inherent in cost data for metal sheet stamping. By evaluating ANNs, we tried to find the best approach for achieving high prediction accuracy and reliability, thereby enhancing the efficiency of our cost estimation model.

Additional research efforts in man-hour prediction across various industries further enrich our understanding. For instance, In [48], authors targeted the power transformers manufacturing sector, comparing Support Vector Machines (SVM), Gaussian Process Regression (GPR), and Adaptive Neural Fuzzy Inference System (ANFIS) models. GPR was found to outperform the others in their dataset, potentially due to its effectiveness in managing noise and uncertainty in production data.

In [49], authors focused on the shipbuilding industry, implementing Multiple Linear Regression (MLR) and Classification and Regression Tree (CART) to predict man-hours in sub-processes. CART outperformed MLR, likely due to its superior handling of categorical and nonlinear data, which are common in such fragmented production processes.

In the construction sector, authors of [50] combined Random Forest (RF) and Linear Regression (LR) to predict Building Information Modeling (BIM) labor costs, finding that the hybrid approach outperformed individual methods. This suggests the potential benefits of methodological hybridization in enhancing prediction accuracy.

These studies collectively highlight the diverse applications and potential of machine learning in cost prediction across industries, informing our approach and methodology in the metal sheet stamping industry, where the challenges of custom, short-run production dominate.

Table 1: Comparison of Existing Studies for Man-hour Prediction of Industrial Products.

Dataset and Paper	SS	# of Features*	Best Model	Year
[37]	150	8	ANN	2014
[49]	300k	11	CART	2015
[51]	99	11	LS-SVM (PSO)	2015
[28]	400k	3D Voxels	CNN	2020
[50]	19	9	RF + LR	2020
[31]	1026	18	RF	2021
[25]	1340	>13	LR	2021
[48]	1249	9	GPR	2022
[52]	>8	4	LS-SVM (PSO)	2023
[53]	1605	10	LR	2023
[1]	4000	47	LightGBM (Optuna)	2023
Our	4890	47 + 3D	LightGBM (Optuna)	2024

SS: Sample Size, **Feature count prior to preprocessing operations*

3. Data

In this research, we utilized two distinct datasets pertaining to the output of sheet metal stamping parts to forecast the operational costs involved in the manufacturing process. These datasets encompass details on the product features as well as the die characteristics necessary for each operation. The primary variable of interest is the work man-hour for each operation, referred to as 'OperationCost.' These datasets were supplied by a sheet metal stamping company.

The supplied product information data set encompasses details regarding each individual product (or part). The data set comprises information on 875 products. It includes the following features:

- **InquiryID**: Distinct identifier assigned to each individual part inquiry.
- **SheetThickness**: Thickness of the metal sheet required for manufacturing the part, represented as a floating-point number.
- **NetX and NetY**: Dimensions of the sheet metal, specified as length and width, respectively.
- **ContourSize**: Size of the contour of the final manufactured part.
- **SurfaceArea**: Total surface area of the completed part.
- **SheetTsMax**: Measure of the tensile strength of the sheet material.
- **SheetElongation**: Attribute describing the elongation capacity of the metal sheet.
- **MetalHardness**: Categorical attribute denoting the hardness of the sheet metal, classified as Soft, Medium, or Hard.
- **Year**: The calendar year in which the inquiry quotation request was received.
- **YearDay**: The specific day of the year, ranging from 0 to 365, on which the inquiry was recorded.

The data set for operations encompasses details related to the attributes of the die and the operations required for the production of each product. The dataset comprises a total of 4000 operations, wherein each product is subjected to between 2 to 8 sequential operations, with certain operations potentially being carried out concurrently. The sub-operations are expressed as natural language strings, which had to be parsed with regex. The dataset features the following attributes:

- **OperationID**: A unique identifier assigned to each row within the operations dataset.
- **InquiryID**: An identifier corresponding to the specific part for which the operation is conducted.
- **OperationOrder**: An indicator of the arrangement of the operation within the manufacturing process.
- **PressTonnage**: The tonnage required in the press during the stamping process.
- **DieX, DieY, DieZ**: The dimensions (length, width, height) of the die.
- **DieWeight**: The weight of the die measured in kilograms.
- **DieFilling**: The percentage of the die's internal space that is filled.
- **Sub-operation features**: Boolean and integer features denoting the presence or frequency of various sub-operations (such as metal sheet blanking, shearing, bending, etc.) and other configurations relevant to the die. The string comprises a series of sub-operation types accompanied by the respective frequency of their execution, with each sub-operation type separated by a comma. Through the use of regular expressions to parse this string, we obtained a frequency list of the sub-operations, which was subsequently integrated into the dataset as die directions T, R, L (booleans indicating the die direction top, right, and left respectively); and sub-operation type execution counts. The sub-operations can be descripbed as:
  - **BLANK**: Cuts out a flat metal piece (blank) from a larger sheet, typically in the shape needed for further forming.
  - **SHEAR**: Cuts or trims metal along a straight or curved line to achieve specific dimensions or separate parts.
  - **BEND**: Deforms the sheet along a straight axis to create angles, folds, or flanges, turning flat sheets into 3D shapes.
  - **DRAW**: Pulls metal into a die cavity to form deep, hollow shapes, commonly used for creating cups or cylindrical parts.
  - **GAUGE**: Measures and controls the thickness of the sheet or part to ensure uniformity and adherence to specifications.
  - **WELD**: Joins two or more metal parts together, typically by applying heat or pressure, to create a single assembly.
  - **PROG (Progressive Stamping)**: Uses multiple stations in a single die to perform a sequence of operations (like blanking, bending, and drawing) on a single part as it moves through the press.
  - **OTHER**: Other infrequent sub-operation types within the dataset, such as coining, ironing, flanging, hemming, embossing, etc., are collected under this type.
- **work man-hour**: The man-hour of operation, serving as the target variable, which we seek to forecast.

For each set of operational data, the corresponding product information from the product dataset is appended. Consequently, this merging of datasets enables machine learning models to predict the work man-hours for each operation. Subsequently, professionals can integrate these costs to establish the target cost for the operations of a given product. There are numerous suboperations, characterized by interconnections among them. Following consultations

with domain experts, suboperations were categorized into primary groups.

Despite the comprehensive nature of the datasets, several data quality issues were identified that could potentially impact model performance. The distribution of the 'OperationCost' variable, our primary target, exhibited notable skewness, predominantly featuring lower values and discrete increments, often in multiples of 50. This pattern suggests possible label noise due to rounding or estimation practices in recording work man-hours. Additionally, the inclusion of operations with costs below 250 and above 3000 introduced potential sampling bias, as these extremes may represent atypical production scenarios or the involvement of subcontractors utilizing different procedures and equipment. Inconsistencies in the numerical representation of sub-operation counts and missing values in certain features necessitated careful data preprocessing and imputation strategies. These biases and noise within the dataset could lead to challenges such as model overfitting or underfitting, adversely affecting the generalization performance of the predictive models. Nonetheless, in this study, these anomalies were retained following preliminary data cleaning, as they are essential for the model to accommodate these atypical samples to ensure robust learning outcomes.

In the previous study [1], the number of samples was lower, which increased after newly processed and provided data samples from the data source. A better way to represent 3D products and the sequential nature of operations can be suggested to increase the performance of the models. Thus, in this study, we added features to represent 3D attributes of the parts. We were able to acquire some features after processing the STL files of each part. These features are volume of the part that is calculated after voxelization of the part, surface area of the part that is calculated directly from the mesh, and number of triangles in the part file (which was correlated with the complexity of the part).

For each sub-operation within the operations, there were inconsistent numerical representations of the step count. This value is incorporated as a new numeric feature when it is available and assigned a value of 0 in its absence. Additionally, the operation dataset is adjusted to include the aggregate count of subsequent and preceding suboperations to provide temporal context to the model. Furthermore, we enhance the whole feature set through the process of feature crossing, which involves the application of multiplicative combinations of pairs of features, as well as the inclusion of squared terms of individual features. Although this method results in an exponential increase in the total number of features, subsequent feature selection is used to mitigate the overall expansion. Specifically, we retain only the features whose importance exceeds the expected importance of the original features.

Given the characteristics of the manufacturing processes, data on certain sub-operation types exhibited imbalances and sparsity in the dataset. This hindered machine learning models from effectively generalizing these operations. To address this issue, we applied the Synthetic Minority Over-sampling Technique (SMOTE) [54] to create synthetic data for these underrepresented sub-operation types. By thoroughly analyzing the dataset's intrinsic patterns and relationships, such as interactions between product dimensions, material properties, and operational parameters, we ensured that the synthetic samples accurately reflected the complexities of real-world metal sheet stamping operations. Subsequently, we employed the Tomek-link method [55] to prune the synthetic data, thereby reducing noise and preventing potential overfitting. This approach not only augmented the dataset's diversity and volume but also led to a significant reduction in the mean absolute error (MAE) for the rare sub-process types, while the other sub-processes exhibited minimal changes. Consequently, the machine learning models were able to learn more generalized and nuanced patterns, improving their predictive performance on unseen data and enhancing their applicability in practical settings.

4. Methods

The data's label values are adjusted by a constant factor to enhance stability during training. Both categorical and numerical features have been reviewed with domain experts and tailored to suit the requirements of each algorithm. Following feature processing, new features are generated based on the data's sequential nature. These novel features encompass information on past and future operations for a single operational step. In the quotation process, experts determine sequential procedures, and similar operations may incur varying costs depending on their position within the sequence.

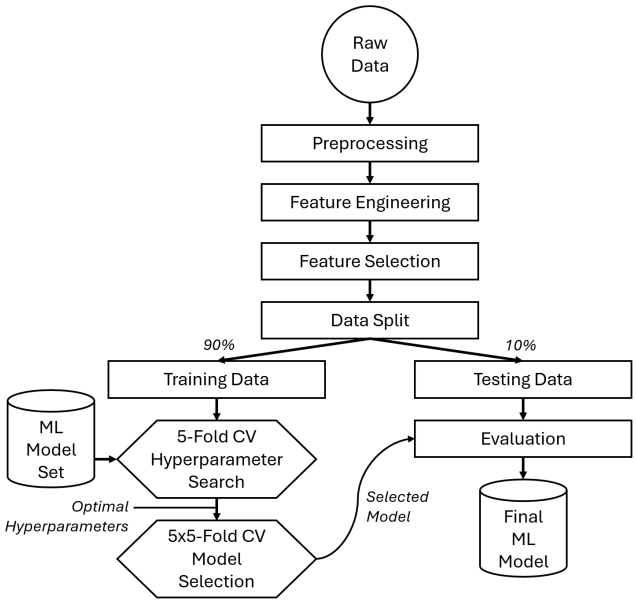


Figure 1: The flow diagram of the process.

Once pre-processing and feature engineering are completed, forward feature selection is employed to reduce the number of features for more stable training. This greedy algorithm in machine learning aims to determine the most relevant features for model prediction. It starts with an empty set and incrementally adds features that most improve model performance until no further significant enhancement is observed or all features are included. Although this method seeks to minimize redundancy and maximize relevance, it can be computationally intensive with high-dimensional datasets and may not always find the optimal subset due to possible local optima.

The data is subsequently divided, with 90% allocated to the training set and 10% to the test set. This test set is used to observe

the capability of the model in unseen data. Each machine learning model undergoes 5-fold cross-validation on the training set to identify the best hyper-parameters. The Optuna library [56] is utilized for hyper-parameter optimization.

To assess the performance of each experimented machine learning model, we conducted 5x5 cross-validation on the training set using the optimal hyper-parameters identified during the hyper-parameter tuning phase. *MAE* and *MAPE* were employed as metrics to compare the various models. Figure 1 provides a flow diagram of the architecture, illustrating how the models are trained and compared.

We conducted trials with a variety of machine learning models. Given their proven effectiveness on tabular datasets, our primary investigation centered on LightGBM [57] as we acquired best results with this method in our previous study [1], while still comparing the method with XGBoost [58]. Additionally, to establish benchmarks, we explored Random Forest, Support Vector Regression, and Multilayer Perceptron techniques. For the models' assessment, we utilized mean absolute error (MAE) and mean absolute percentage error (*MAPE*). MAE quantifies the average absolute deviation between the forecasted and observed values, whereas *MAPE* calculates the average percentage deviation between the predicted and true values.

To operationalize the predictive models developed in this study, we implemented a 3D shape-based pricing service designed to integrate seamlessly with the company's existing quotation system. This service provides a machine learning-based tool for predicting industrial product prices, specifically focusing on work-hour estimation for labor costs. It accepts inputs such as 3D models in STL format and various numerical and categorical parameters; including material thickness, type, surface area, hardness, and operation-specific details like mold dimensions and press types, all of which are elaborated in the data section.

Users interact with the service via application programming interface (API) which may be augmented into a dedicated user interface, permitting manual data input or the selection of pre-existing components from the system. The API also allows users to enable or disable the inclusion of 3D data in the predictions. Upon receiving input, the system extracts key features from the 3D model, such as triangle count, total surface area, and volume. These features, along with additional parameters, are fed into the selected machine learning model.

The predicted work hours are converted into a labor cost using a configurable multiplier, allowing the cost to be adjusted based on departmental rates or project-specific requirements. The system also calculates department-specific costs proportionally to the work hours, providing a detailed cost breakdown for activities such as CAD, CAM, 2D cutting, drilling & machining, assembly, measurement, and various CNC processes. This flexible approach enables users to make informed pricing decisions quickly, streamlining the cost estimation process.

For deployment and integration, the service is containerized using Docker and designed to run on-premises to ensure data confidentiality. It exposes a set of API endpoints for various functionalities, including data operations, model training, and prediction. These APIs allow for uploading 3D models and tabular data, configuring cost ratios, training new models, and making predictions. The service supports both single models trained on the entire dataset and ensemble models trained using 5-fold cross-validation, providing options for balancing performance and computational efficiency. This modular and secure design facilitates easy integration with other applications and supports the scalability of the solution within the company's infrastructure.

5. Results and Discussion

In this section, we present a comprehensive analysis of our findings. We begin with an overview of the experimental setup used for hyperparameter tuning, followed by a detailed examination of model experimentation results. Subsequently, we delve into model interpretability through the utilization of SHAP values. Finally, we examine the results of software testing, with particular emphasis on usability and performance metrics.

5.1. Experimental Setup

In our experiments with machine learning models, we meticulously optimized the hyperparameters to enhance the models' performance in predicting work man-hours for metal sheet stamping projects. The hyperparameter tuning was conducted using the Tree-structured Parzen Estimator available in the Optuna library [56], which efficiently explores the hyperparameter space to identify optimal settings.

An investigation of the final acquiredhyperparemeters of a model, LightGBM, can provide a more profound comprehension of the obtained results. The final model employed the 'dart' boosting type, which integrates dropout techniques into the boosting process to prevent overfitting by randomly dropping trees during training. We selected a learning rate of 0.33 to accelerate convergence, allowing the model to learn efficiently from the data without excessively prolonging the training time. A maximum depth of 30 and a high number of leaves (208) were set to enable the model to capture complex nonlinear relationships inherent in the manufacturing data, accommodating the intricate interactions among numerous features.

Minimal regularization was applied, with λ_{l1}, λ_{l2} set to near-zero values (1.06×10^{-8} and 2.97×10^{-4}, respectively), indicating that strong regularization was unnecessary due to effective overfitting control by other parameters like feature and bagging fractions. To introduce randomness and promote generalization, we utilized a feature fraction of 0.5675 and a bagging fraction of 0.84 with a bagging frequency of 2. This ensured that each iteration trained on a random subset of features and data samples, reducing the risk of the model becoming too tailored to specific patterns in the training set.

We set the minimum data in a leaf to 1, allowing the model to capture rare patterns and exceptions in the data, which is crucial for accurately predicting work man hour costs associated with infrequent sub-operation types. The maximum bin was configured to 212, permitting finer discretization of continuous features and enabling the model to capture subtle variations in feature values that significantly impact the target variable. By leveraging the Optuna library's Tree-structured Parzen Estimator for hyperparameter optimization, we were able to systematically explore the hyperparameter space and identify the optimal settings that maximized the model's

predictive performance. This careful tuning was essential for developing a robust model capable of achieving high predictive accuracy and meeting our target error rate, thereby effectively supporting decision-making processes in the manufacturing workflow.

5.2. Model Experimentation Results

We utilized the test set and implemented 5x5 cross-validation on the training set to evaluate various models. For the assessment, Mean Absolute Error (*MAE*) and Mean Absolute Percentage Error (*MAPE*) metrics were chosen due to their interpretability and wide acceptance within the domain. The outcomes of these evaluations are presented in Table 2. As expected, LightGBM and XGBoost exhibited superior performance compared to other models, with LightGBM achieving the lowest *MAE* and *MAPE* values during cross-validation (CV). The results on the test data from the top-performing model (LightGBM) are depicted in Figure 2. Based on consultations with industry experts, a model must exhibit a maximum *MAPE* of 10% to be deemed valuable, which constitutes the target Key Performance Indicator (KPI) for this study. The findings suggest that the majority of samples fall within this acceptable range. Furthermode, the models were compared using a variety of KPI metrics. Figure 3 depicts the proportion of samples accurately predicted according to these selected KPI metrics across all experiments. The selected 10% error threshold KPI target is also indicated in the Figure 3 as a red dashed line. For the top-selected model, it is apparent that 75% of the samples are within this acceptable range. Therefore, given the complexity of the task, we conclude that it is feasible to develop and deploy models for predicting work man hour costs in the sheet metal stamping industry.

Table 2: Comparison table of the cross validation results

Models	Results	
	5x5 CV MAPE	***5x5 CV MAE***
LightGBM*	10.89%	71.49
LightGBM	**10.78%**	**70.37**
XgBoost*	11.30%	73.72
XgBoost	11.23%	72.25
KNN Regressor	20.55%	122.01
MLP	16.61%	105.25
Linear Regression	26.35%	136.99

Models that end with "*" indicates that the model is trained without the additional 3D data features.

We further assessed how feature engineering techniques affect model performance. Our findings revealed that performance notably improves for both LightGBM and XgBoost after applying feature selection and generating synthetic data for chosen features. It is important to highlight that cost prediction in any field may be constrained by data representation limitations. The features depicting the product and operations might be overly generic, lacking detailed representation for each training example. Additionally, given the typical scarcity of industrial data in cost estimation tasks, additional efforts could focus on data augmentation. This could unlock the potential of more sophisticated machine learning models, such as Transformers and DNNs.

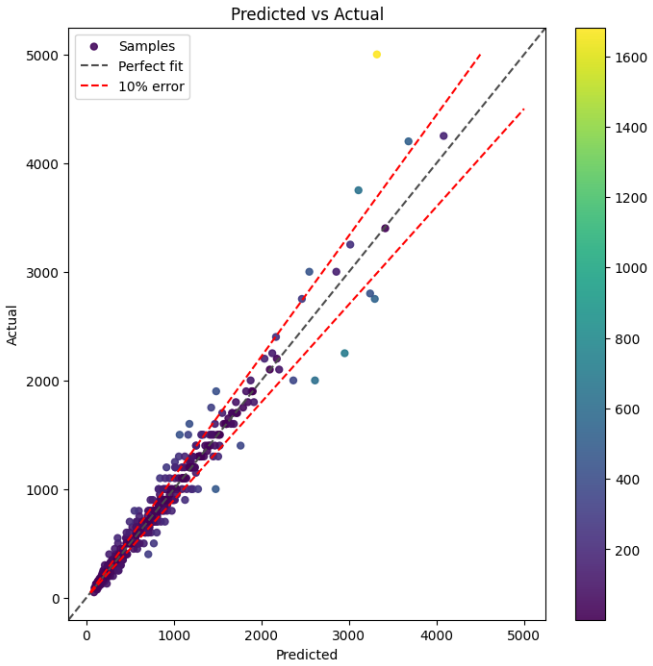


Figure 2: Evaluation of the best model on the test set.

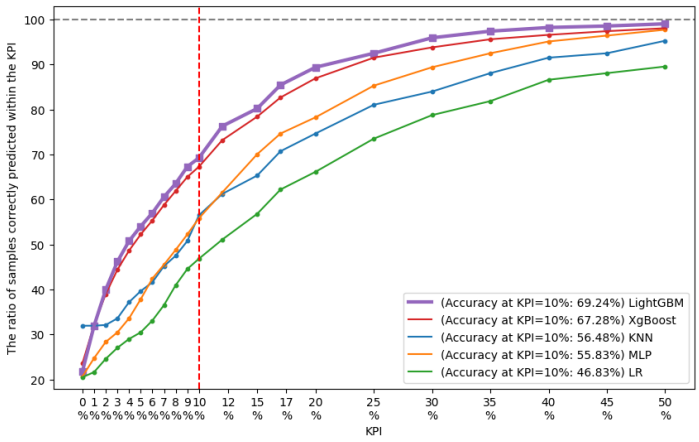


Figure 3: Change of the ratio of samples correctly predicted within different KPI metrics.

5.3. Model Interpretability

In addition, to gain deeper insights into the model's decision-making process, we employ SHAP (SHapley Additive exPlanations) values [59] to interpret the contribution of each feature to the predictions. SHAP is a model-agnostic interpretability method based on cooperative game theory, which assigns each feature an importance value by calculating its average marginal contribution across all possible feature combinations. By analyzing the SHAP values for our model, we found that press tonnage, the presence of a progressive operation, and die dimensions (DieX, DieY, DieZ) significantly influence the work man-hour estimation, as depicted in Figure 4. These features

have the highest SHAP values, indicating they contribute most to the predicted costs.

Specifically, higher press tonnage is associated with increased work man-hours, aligning with domain knowledge that higher tonnage presses require more setup time and operational complexity. The presence of a progressive operation also contributes to higher predicted man-hours due to the additional tooling and coordination required for such operations. Larger die dimensions (length, width, height) impact the prediction by indicating more substantial or complex dies, which typically necessitate more labor for handling and setup. We also observed that the high triangle count of the part 3D model, which correlates with the high part complexity, increases the work man-hours in general.

Other features, such as material properties and other sub-operation counts, have a comparatively moderate effect on the prediction. The SHAP analysis enhances the interpretability of our model by illustrating how each feature influences the output, ensuring that the model's behavior aligns with expert understanding. This transparency in the decision-making process not only validates the model's reliability but also builds trust with stakeholders by demonstrating that the predictions are based on logical and explainable factors relevant to the manufacturing context.

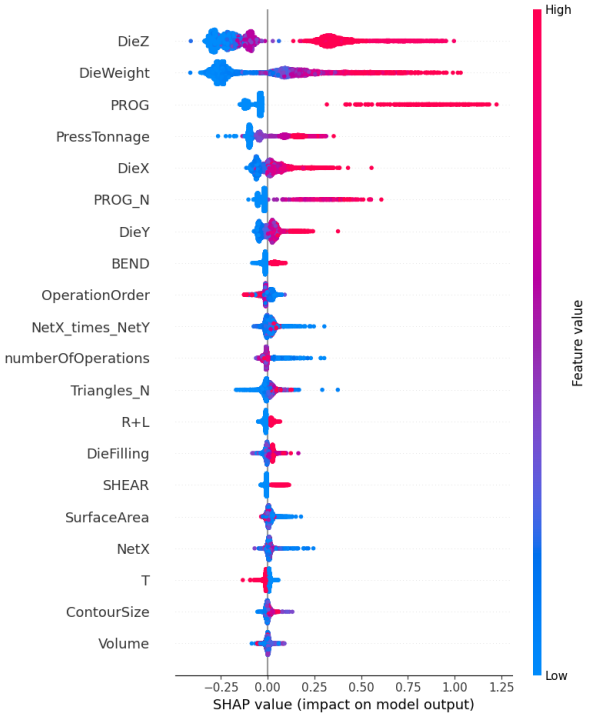


Figure 4: The SHAPley summarization of the features.

5.4. Software Test Results

In our implementation of the pricing service, LightGBM was selected due to its superior performance in predicting work hours, achieving a target error rate of less than or equal to 10%. The adoption of this machine learning model has significantly reduced the time required to generate quotations. On average, the inference time of the model is less than 15 seconds. While the analysis of the 3D part can add some time, especially if the data is not already stored in the database, resulting in a total average inference time of approximately 2 minutes ± 15 seconds, the overall process still represents a substantial improvement. Considering the reported time from the manufacturer, this approach reduces the time taken to respond to customer inquiries by 90%, which is crucial in industries where speed is a competitive advantage. This significant reduction in response time not only enhances operational efficiency but also provides a competitive edge in the fast-paced metal sheet stamping industry.

6. Conclusions

In this study, we extended our previous study on work-man hour forecasting in metal sheet stamping processes by conducting a comparative assessment of the efficiency of diverse machine learning algorithms. Additionally, the research investigates the influence of different feature engineering strategies on the results. This problem is formulated and analyzed within the framework of a regression model. We identified the most influential variables and features affecting work man-hours within the field. Additionally, we examined the performance of the models and outlined current limitations that require further investigation. The findings indicated that LightGBM and XGBoost achieved the highest accuracy (lowest *MAPE* error of 10.78%) compared to other experimented models. exhibiting commendable performance. While the initial study improved the predictive performance of the models through feature selection and synthetic data generation techniques, the present study focused on augmenting the dataset with additional real-world data and incorporating advanced feature engineering methods. With the additional improvements, most influential variables contributing to the work man-hour was similar to the previous study, as the the die dimensions, the amount of press tonnage, and the presence of progressive operations, with the exception of die weight. Collaboration with domain experts proved instrumental in understanding the utilization of certain features and the overall constraints of the project. Overall, our research underscores the potential of machine learning models in the context of work man-hours for metal sheet stamping projects and emphasizes the importance of feature engineering and the incorporation of domain-specific knowledge in enhancing model performance. The main limitation of this research is the insufficient availability of real-world data, which obstructs the application of deep learning techniques that could more effectively utilize the 3D models. Future research could focus on improved data representation methods using the 3D part data, such as image renders of the 3D part. Additionally, further research may explore the deployment of deep learning methods, that are adept at leveraging voxel-based 3D data.

Conflict of Interest The authors declare that they have no conflict of interest.

Acknowledgment This work was supported by the TÜBTAK TEYDEB Program with Project no: 9210055. The dataset and the use case problem were provided by ERMETAL A.Ş. The authors thank Cem Yıldız and Ali Erman Erten for sharing their expertise with us.

References

[1] A. E. Ünal, H. Boyar, B. Kuleli Pak, C. Yıldız, A. E. Erten, V. C. Güngör, "Man-hour Prediction for Complex Industrial Products," 4th International Informatics and Software Engineering Conference, 2023.

[2] T. Yeh, S. Deng, "Application of machine learning methods to cost estimation of product life cycle," International Journal of Computer Integrated Manufacturing, **25**(4-5), 340–352, 2012, doi:10.1080/0951192x.2011.645381.

[3] H. Salleh, "Selecting a standard set of attributes for the development of machine learning models of building project cost estimation," Planning Malaysia, **21**, 2023, doi:10.21837/pm.v21i29.1359.

[4] S. Yoo, N. Kang, "Explainable artificial intelligence for manufacturing cost estimation and machining feature visualization," Expert Systems With Applications, **183**, 115430, 2021, doi:10.1016/j.eswa.2021.115430.

[5] F. Ning, Y. Shi, M. Cai, W. Xu, X. Zhang, "Manufacturing cost estimation based on a deep-learning method," Journal of Manufacturing Systems, **54**, 186–195, 2020, doi:10.1016/j.jmsy.2019.12.005.

[6] O. Kurasova, V. Marcinkevičius, V. Medvedev, B. Mikulskienė, "Early cost estimation in customized furniture manufacturing using machine learning," International Journal of Machine Learning and Computing, **11**(1), 28–33, 2021, doi:10.18178/ijmlc.2021.11.1.1010.

[7] L. Yang, J. Li, F. Chao, P. Hackney, M. Flanagan, "Job shop planning and scheduling for manufacturers with manual operations," Expert Systems, **38**(7), 2018, doi:10.1111/exsy.12315.

[8] F. Bodendorf, J. Franke, "Application of the technology acceptance model to an intelligent cost estimation system: an empirical study in the automotive industry," 2022, doi:10.24251/hicss.2022.144.

[9] M. Atia, J. Khalil, M. Mokhtar, "A cost estimation model for machining operations; an ann parametric approach," Journal of Al-Azhar University Engineering Sector, **12**(44), 878–885, 2017, doi:10.21608/auej.2017.19195.

[10] J. Loyer, E. Henriques, M. Fontul, S. Wiseall, "Comparison of machine learning methods applied to the estimation of manufacturing cost of jet engine components," International Journal of Production Economics, **178**, 109–119, 2016, doi:10.1016/j.ijpe.2016.05.006.

[11] F. Silva, V. Sousa, A. Pinto, L. Ferreira, M. Pereira, "Build-up an economical tool for machining operations cost estimation," Metals, **12**(7), 1205, 2022, doi:10.3390/met12071205.

[12] K. Manjunath, S. Tewary, N. Khatri, K. Cheng, "Monitoring and predicting the surface generation and surface roughness in ultraprecision machining: a critical review," Machines, **9**(12), 369, 2021, doi:10.3390/machines9120369.

[13] H. Y. Gong, J. Y. Wang, Z. H. Zhao, "Study on the springback characteristics of cr340la steel during the typical auto part stamping process," Advanced Materials Research, **322**, 98–101, 2011, doi:10.4028/www.scientific.net/amr.322.98.

[14] R. Zeng, L. Huang, J. Li, "Fracture prediction in sheet metal stamping based on a modified ductile fracture criterion," Key Engineering Materials, **639**, 543–550, 2015, doi:10.4028/www.scientific.net/kem.639.543.

[15] S. Zvonov, Y. Klochkov, "Computer-aided modelling of a latch die cutting in deform - 2d software system," Key Engineering Materials, **685**, 811–815, 2016, doi:10.4028/www.scientific.net/kem.685.811.

[16] S. Kokare, "Toward cleaner space explorations: a comparative life cycle assessment of spacecraft propeller tank manufacturing technologies," The International Journal of Advanced Manufacturing Technology, **133**(1-2), 369–389, 2024, doi:10.1007/s00170-024-13745-y.

[17] M. Moghadam, C. Nielsen, N. Bay, "Analysis of the risk of galling in sheet metal stamping dies with drawbeads," Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, **234**(9), 1207–1214, 2020, doi:10.1177/0954405420911307.

[18] T. Chen, "Xgboost: a scalable tree boosting system," 2016, doi:10.48550/arxiv.1603.02754.

[19] F. Wang, "Extended-window algorithms for model prediction applied to hybrid power systems," Technologies, **12**(1), 6, 2024, doi:10.3390/technologies12010006.

[20] S. Guo, "Revolutionizing the used car market: predicting prices with xgboost," Applied and Computational Engineering, **48**(1), 173–180, 2024, doi:10.54254/2755-2721/48/20241349.

[21] T. Duan, A. Avati, D. Ding, K. Thai, S. Basu, A. Ng, et al., "Ngboost: natural gradient boosting for probabilistic prediction," 2019, doi:10.48550/arxiv.1910.03225.

[22] D. Ruta, M. Liu, L. Cen, Q. Vu, "Diversified gradient boosting ensembles for prediction of the cost of forwarding contracts," 2022, doi:10.15439/2022f291.

[23] S. Touzani, J. Granderson, S. Fernandes, "Gradient boosting machine for modeling the energy consumption of commercial buildings," Energy and Buildings, **158**, 1533–1543, 2018, doi:10.1016/j.enbuild.2017.11.039.

[24] P. Panjee, "A generalized linear model and machine learning approach for predicting the frequency and severity of cargo insurance in Thailand's border trade context," Risks, **12**(2), 25, 2024, doi:10.3390/risks12020025.

[25] F. Bodendorf, J. Franke, "A machine learning approach to estimate product costs in the early product design phase: a use case from the automotive industry," Procedia CIRP, **100**, 643–648, 2021.

[26] L. Liu, E. Chen, Y. Ding, "Tr-net: a transformer-based neural network for point cloud processing," Machines, **10**(7), 517, 2022, doi:10.3390/machines10070517.

[27] Z. Yang, Y. Sun, S. Liu, X. Shen, J. Jia, "Std: sparse-to-dense 3d object detector for point cloud," in Proceedings of the IEEE International Conference on Computer Vision, 204, 2019, doi:10.1109/iccv.2019.00204.

[28] F. Ning, Y. Shi, M. Cai, W. Xu, X. Zhang, "Manufacturing cost estimation based on a deep-learning method," Journal of Manufacturing Systems, **54**, 186–195, 2020.

[29] X. Zhang, "Multiattention mechanism 3d object detection algorithm based on rgb and lidar fusion for intelligent driving," Sensors, **23**(21), 8732, 2023, doi:10.3390/s23218732.

[30] B. Huang, Y. Feng, T. Liang, "A voxel generator based on autoencoder," Applied Sciences, **12**(21), 10757, 2022, doi:10.3390/app122110757.

[31] O. Kurasova, V. Marcinkevičius, V. Medvedev, B. Mikulskienė, "Early cost estimation in customized furniture manufacturing using machine learning," International journal of machine learning and computing, **11**(1), 28–33, 2021.

[32] V. Svetnik, A. Liaw, C. Tong, J. Culberson, R. Sheridan, B. Feuston, "Random forest: a classification and regression tool for compound classification and qsar modeling," Journal of Chemical Information and Computer Sciences, **43**(6), 1947–1958, 2003, doi:10.1021/ci034160g.

[33] H. Zermane, A. Drardja, "Development of an efficient cement production monitoring system based on the improved random forest algorithm," The International Journal of Advanced Manufacturing Technology, **120**(3-4), 1853–1866, 2022, doi:10.1007/s00170-022-08884-z.

[34] G. Pan, "Xgboost and random forest algorithm for supply fraud forecasting," 2022, doi:10.1117/12.2641948.

[35] C. Strobl, A. Boulesteix, T. Kneib, T. Augustin, A. Zeileis, "Conditional variable importance for random forests," BMC Bioinformatics, **9**(1), 2008, doi:10.1186/1471-2105-9-307.

[36] A. Ziegler, I. König, "Mining data with random forests: current options for real-world applications," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, **4**(1), 55–63, 2013, doi:10.1002/widm.1114.

[37] B. Özcan, A. Fığlalı, "Artificial neural networks for the cost estimation of stamping dies," Neural computing and applications, **25**, 717–726, 2014.

[38] Z. Leszczyński, T. Jasiński, "An artificial neural networks approach to product cost estimation: the case study for electric motor," Informatyka Ekonomiczna, **1**(47), 72–84, 2018, doi:10.15611/ie.2018.1.06.

[39] B. Waziri, K. Bala, S. Bustani, "Artificial neural networks in construction engineering and management," International Journal of Architecture Engineering and Construction, **6**(1), 2017, doi:10.7492/ijaec.2017.006.

[40] M. Meharie, W. Mengesha, Z. Gariy, R. Mutuku, "Application of stacking ensemble machine learning algorithm in predicting the cost of highway construction projects," Engineering Construction & Architectural Management, **29**(7), 2836–2853, 2021, doi:10.1108/ecam-02-2020-0128.

[41] I. Peško, V. Mučenski, M. Šešlija, N. Radović, A. Vujkov, D. Bibić, et al., "Estimation of costs and durations of construction of urban roads using ann and svm," Complexity, 1–13, 2017, doi:10.1155/2017/2450370.

[42] S. Magdum, A. Adamuthe, "Construction cost prediction using neural networks," Ictact Journal on Soft Computing, **8**(1), 1549–1556, 2017, doi:10.21917/ijsc.2017.0216.

[43] O. Durán, N. Rodríguez, L. Consalter, "Neural networks for cost estimation of shell and tube heat exchangers," Expert Systems With Applications, **36**(4), 7435–7440, 2009, doi:10.1016/j.eswa.2008.09.014.

[44] M. Bouabaz, M. Hamami, "A cost estimation model for repair bridges based on artificial neural network," American Journal of Applied Sciences, **5**(4), 334–339, 2008, doi:10.3844/ajassp.2008.334.339.

[45] K. Kim, I. Han, "Application of a hybrid genetic algorithm and neural network approach in activity-based costing," Expert Systems With Applications, **24**(1), 73–77, 2003, doi:10.1016/s0957-4174(02)00084-2.

[46] M. Juszczyk, "Early fast cost estimates of sewerage projects construction costs based on ensembles of neural networks," Applied Sciences, **13**(23), 12744, 2023, doi:10.3390/app132312744.

[47] A. Zouidi, F. Fnaiech, K. Al-Haddad, "A multi-layer neural network and an adaptive linear combiner for on-line harmonic tracking," 2007, doi:10.1109/wisp.2007.4447612.

[48] K. Işık, S. E. Alptekin, "A benchmark comparison of Gaussian process regression, support vector machines, and ANFIS for man-hour prediction in power transformers manufacturing," Procedia Computer Science, **207**, 2567–2577, 2022.

[49] M. Hur, S. K. Lee, B. Kim, S. Cho, D. Lee, D. Lee, "A study on the man-hour prediction system for shipbuilding," Journal of Intelligent Manufacturing, **26**, 1267–1279, 2015.

[50] C. H. Huang, S. H. Hsieh, "Predicting BIM labor cost with random forest and simple linear regression," Automation in Construction, **118**, 103280, 2020.

[51] T. Yu, H. Cai, "The prediction of the man-hour in aircraft assembly based on support vector machine particle swarm optimization," Journal of Aerospace Technology and Management, **7**, 19–30, 2015.

[52] S. Guo, T. Jiang, "Cost prediction of equipment system using LS-SVM with PSO," in 2007 International Conference on Wireless Communications, Networking and Mobile Computing, 5285–5288, IEEE, 2007.

[53] X. Hu, M. Lu, S. AbouRizk, "BIM-based data mining approach to estimating job man-hour requirements in structural steel fabrication," in Proceedings of the Winter Simulation Conference 2014, 3399–3410, IEEE, 2014.

[54] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, **16**, 321–357, 2002.

[55] R. M. Pereira, Y. M. Costa, C. N. Silla Jr., "MLTL: A multi-label approach for the Tomek Link undersampling algorithm," Neurocomputing, **383**, 95–105, 2020, doi:https://doi.org/10.1016/j.neucom.2019.11.076.

[56] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2623–2631, 2019.

[57] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, et al., "Lightgbm: A highly efficient gradient boosting decision tree," Advances in neural information processing systems, **30**, 2017.

[58] T. Chen, C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 785–794, 2016.

[59] S. M. Lundberg, S. I. Lee, "A unified approach to interpreting model predictions," Advances in neural information processing systems, **30**, 2017.

Advances in Science, Technology and Engineering Systems Journal Vol. 9, No. 6, 12-20 (2024)

www.astesj.com

Special Issue on Innovation In Computing, Engineering Science & Technology

ASTESJ

ISSN: 2415-6698

Advanced Fall Analysis for Elderly Monitoring Using Feature Fusion and CNN-LSTM: A Multi-Camera Approach

Win Pa Pa San[1], Myo Khaing[2]

[1]*Image and Signal Processing Lab, University of Computer Studies, Mandalay, Mandalay, 05071, Myanmar*

[2]*Faculty of Computer Science, University of Computer Studies, Mandalay, Mandalay, 05071, Myanmar*

ARTICLE INFO

Article history:
Received: 15th September, 2024
Revised: 30th September, 2024
Accepted: 15th October, 2024
Online: 30th November, 2024

Keywords:
Feature Fusion
Human Silhouette Image (HSI)
Silhouette History Images (SHI)
Dense Optical Flow (DOF)
Convolutional Neural Network (CNN)
Long Short-Term Memory (LSTM)

ABSTRACT

As society ages, the imbalance between family caregivers and elderly individuals increases, leading to inadequate support for seniors in many regions. This situation has ignited interest in automatic health monitoring systems, particularly in fall detection, due to the significant health risks that falls pose to older adults. This research presents a vision-based fall detection system that employs computer vision and deep learning to improve elderly care. Traditional systems often struggle to accurately detect falls from various camera angles, as they typically rely on static assessments of body posture. To tackle this challenge, we implement a feature fusion strategy within a deep learning framework to enhance detection accuracy across diverse perspectives. The process begins by generating a Human Silhouette Image (HSI) through background subtraction. By combining silhouette images from two consecutive frames, we create a Silhouette History Image (SHI), which captures the shape features of the individual. Simultaneously, Dense Optical Flow (DOF) extracts motion features from the same frames, allowing us to merge these with the SHI for a comprehensive input image. This fused representation is then processed using a pre-trained Convolutional Neural Network (CNN) to extract deep features. A Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) is subsequently trained on these features to recognize patterns indicative of fall events. Our approach's effectiveness is validated through experiments on the UP-fall detection dataset, which includes 1,122 action videos and achieves an impressive 99% accuracy in fall detection.

1. Introduction

The aging population is rapidly growing worldwide, leading to a significant increase in the number of elderly individuals who require constant care and monitoring. As a result, the ratio of family caregivers to elderly individuals is becoming increasingly unbalanced, especially in countries with higher life expectancies. This imbalance has created a pressing need for automatic health monitoring systems that can provide timely and efficient care for the elderly. One of the most critical aspects of such health monitoring systems is the detection of falls, a leading cause of injury and hospitalization among older adults.

Falls among the elderly can occur for various reasons, including heart attacks, high blood pressure, and other home accidents. The consequences of falls can be severe, often leading to a decline in physical and mental health, reduced mobility, and increased dependence on caregivers. Therefore, accurately detecting falls in real-time is essential for preventing further injuries and ensuring prompt medical attention. Despite the importance of fall detection, traditional vision-based systems face significant challenges in achieving reliable performance across different environments and camera viewpoints.

In recent years, computer vision and machine learning have paved the way for more sophisticated fall detection systems. Convolutional Neural Networks (CNNs) have shown remarkable success in various image processing and object recognition tasks, making them suitable candidates for analyzing video data in fall detection applications. However, static image-based approaches often struggle to capture the temporal dynamics of fall events, which are crucial for accurate detection. This limitation can be addressed by integrating Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, which excel at learning temporal dependencies in sequential data.

* Win Pa Pa San, University of Computer Studies, Mandalay, Mandalay, 05071, Myanmar, +959262988945, winpapasan@ucsm.edu.mm

The proposed fall detection system leverages the strengths of both CNNs and LSTMs, combined with a feature fusion approach to enhance the accuracy and robustness of fall detection. The system utilizes multiple cameras to capture different viewpoints of the monitoring area, providing a comprehensive view of the scene. Human silhouette images are extracted from two consecutive video frames and fused into a Silhouette History Image (SHI), which serves as a shape feature representing the subject's posture over time. Additionally, Dense Optical Flow (DOF) is computed to capture motion features between frames, offering valuable information about the subject's movements.

By fusing SHI and DOF features, the system creates a rich representation of both spatial and temporal aspects of the scene. These fused features are then fed into a pre-trained CNN to extract deep features, which are subsequently processed by an LSTM network to recognize fall events. The use of multiple cameras ensures that the system can detect falls from various angles, overcoming the limitations of single-camera setups. Furthermore, the feature fusion approach enables the system to capture subtle changes in posture and movement, improving the overall detection accuracy.

To evaluate the effectiveness of the proposed system, experiments were conducted using the publicly available UP-Fall detection dataset. The results demonstrate that the proposed method outperforms traditional vision-based fall detection systems, achieving superior performance in terms of accuracy and robustness. This research highlights the potential of combining feature fusion with CNN-LSTM architectures for developing advanced fall detection systems that can significantly enhance the safety and well-being of elderly individuals.

The primary aim of this research is to develop an advanced fall detection system that accurately identifies fall events in real-time, leveraging feature fusion and CNN-LSTM architectures within a multi-camera setup. The specific objectives are:

To design a robust fall detection framework that integrates shape and motion features using Silhouette History Images (SHI) and Dense Optical Flow (DOF).

- To employ a pre-trained CNN for deep feature extraction and an LSTM network for temporal sequence analysis to improve fall detection accuracy.
- To validate the effectiveness of the proposed system through extensive experiments using a publicly available dataset, ensuring its practical applicability in various indoor environments.

The motivation for this research stems from the growing need for reliable and efficient fall detection systems in elderly care. With the increasing elderly population, there is a heightened demand for solutions that can monitor and ensure the safety of older adults, particularly those living alone or in assisted living facilities. Existing fall detection systems often struggle with accuracy due to limitations in capturing dynamic movements and variations in camera viewpoints. By addressing these challenges through the integration of advanced machine learning techniques and a multi-camera approach, this research aims to provide a more dependable solution that enhances the quality of life for the elderly.

Traditional vision-based fall detection systems face several challenges, including:

- Inability to capture temporal dynamics of fall events, leading to missed detections or false alarms.
- Limited performance due to reliance on single-camera setups, which cannot cover all angles and may result in occlusions.
- Difficulty in accurately distinguishing between falls and other similar activities, such as sitting down abruptly.

The proposed system combines CNN and LSTM networks to leverage their strengths in spatial and temporal feature extraction. The use of multiple cameras ensures comprehensive coverage of the monitored area, reducing the likelihood of occlusions and improving detection reliability. Feature fusion of SHI and DOF provides a rich representation of both posture and movement, enabling the system to differentiate between falls and non-fall activities more accurately.

This research makes several key contributions to the field of fall detection:

- Introduction of a novel feature fusion approach that combines SHI and DOF to capture both shape and motion characteristics of potential fall events.
- Development of a hybrid CNN-LSTM architecture that effectively integrates spatial and temporal features for enhanced fall detection performance.
- Implementation of a multi-camera system that overcomes the limitations of single-camera setups, providing a more robust and reliable solution for real-world applications.
- Extensive experimental validation using the UP-Fall detection dataset, demonstrating the superior accuracy and robustness of the proposed method compared to traditional systems.

By addressing the limitations of existing fall detection approaches and introducing innovative solutions, this research contributes to the advancement of health monitoring technologies, ultimately improving the safety and well-being of elderly individuals. Moreover, the proposed system can be applied to a smart home system to assist and provide telehealth services for the elderly.

This paper is organized as follows. Section I describes the objectives, motivations, system problem with solution, and contribution of this study. The literature survey about various fall detections is analyzed in Section II. The system overview and the detailed explanation of this study are presented and the experimental results and comparison with the results of the other existing methods are presented in Section III. Some discussion about the pros and cons of the proposed system are discussed in Section IV. Finally, the conclusion and future work are drawn in Section V.

2. Related Work

The advancement of sophisticated sensors and devices has captured the interest of many researchers focused on artificial intelligence systems. This is particularly true for applications such

as smart home systems, patient monitoring, surveillance, and elderly monitoring, where various sensor-based and camera-based approaches have been proposed. Fall detection systems, in particular, can be classified into two categories based on the sensors used: sensor-based and camera (vision)-based.

2.1. Sensor-based Fall detection

Fall detection sensors typically incorporate accelerometers and gyroscopes to monitor the acceleration and orientation of elderly individuals. When attached to various body parts, accelerometers collect acceleration data during falls. One proposed system [1] employs accelerometers and gyroscopes mounted on the gait to assess balance, detect falls, and evaluate fall risk. In a different approach, Lindeman et al. integrated accelerometer sensors into a hearing aid positioned behind the ear [2]. Another fall detection system [3] identifies falls and locates the fallen individual. This system utilizes a sensor attached to the waist to detect backward and sideways falls based on the wearer's final orientation.

Additionally, the authors in [4] developed a machine learning-based fall detection system that utilizes temporal and magnitude features extracted from acceleration signals. These features were used to train a Support Vector Machine classifier for fall identification. Bianchi et al. implemented a fall detection system using barometric pressure sensors, evaluating its performance against accelerometer-based systems; this system classifies falls based on postural orientation and altitude changes [5]. In [6], another system was proposed that not only detects falls but also assesses injury severity, employing multiple accelerometers attached to joints to analyze three-axis acceleration data. Furthermore, in [7], the authors introduced a fall detection system that combines accelerometer sensors with the Discrete Wavelet Transform (DWT) and Support Vector Machine (SVM) algorithm.

2.2. Vision-based Fall detection

Numerous fall detection systems have been developed in recent years, each utilizing different techniques to enhance accuracy and reliability. A notable approach employs key points of the human skeleton detected via OpenPose, as demonstrated in [8]. This system identifies falls based on the speed of descent of the hip joint, the centerline angle, and the body's width-to-height ratio. While it achieves 98.3% sensitivity, 95% specificity, and 97% accuracy on a dataset of 60 falling and 40 non-falling actions, the system encounters challenges with partial occlusion and recognizing falls from multiple directions.

Another vision-based approach for fall detection, utilizing multiple cameras and convolutional neural networks (CNNs), was proposed in [9]. This system leverages optical flow to capture relative motion between consecutive images and trains three CNN models to process visual features from different camera angles. The results on the UP-Fall detection dataset demonstrated 95.64% accuracy, 97.95% sensitivity, and 83.08% specificity. However, the system's performance is impacted by environmental changes and occlusions. In [10], the authors developed a fall detection system that employs features extracted by Inception v3 and a MobileNet model for human detection. By applying transfer learning, they achieved 98.5% accuracy, 97.2% specificity, and 93.47% sensitivity on the FDD dataset, and 91.5% accuracy, 94% specificity, and 100% sensitivity on the URFD dataset. Nonetheless, managing occlusions continues to pose a significant challenge.

Similarly, in [11], the authors proposed a vision-based fall detection method using CNNs, which involved a three-step training process: initial training with ImageNet, motion modeling with UCF101, and fine-tuning specifically for fall detection. Testing on the URFD, Multicam, and FDD datasets resulted in accuracy rates of 95%, 96%, and 97%, respectively. While the results are promising, the system requires improvements in avoiding image preprocessing issues and managing occlusions and multi-person detection. In [12], the authors combined histograms of oriented gradients (HOG), local binary patterns (LBP), and Caffe features for fall detection. Their system utilized VIBE+ for human detection and extraction, along with SVM for classification, achieving sensitivities of 95%, 93.3%, and 92.9%, and specificities of 97.5%, 92.2%, and 86.4% on the Multicam, Chua's dataset, and their dataset, respectively. However, handling occlusions remains a challenge.

Furthermore, in [13], the authors focused on detecting fallen individuals using assistive robots. Their system utilized features such as the aspect ratio of the bounding box, normalized bounding box width, and bottom coordinate, employing an SVM-based classifier. Testing on the FPDS dataset yielded 100% precision and 99.74% recall. However, the system requires enhancements in occlusion detection and minimizing image preprocessing issues.

These studies underscore several common challenges fall detection systems face, including occlusion handling, adaptability to diverse environmental conditions, effective feature extraction and fusion, thorough testing across varied datasets, and detecting falls in multi-person environments. The proposed advanced fall detection system aims to tackle these issues by integrating shape and motion features, utilizing a hybrid CNN-LSTM architecture, and employing a multi-camera setup. This approach promises to enhance the accuracy and reliability of fall detection, making significant progress toward robust and practical real-world applications.

3. Material and Methods

The purpose system flow of the block diagram illustrating the system flow is shown in Figure. 1 of the Advanced Fall Detection System Using Feature Fusion and CNN-LSTM. They are:

- Video Input: Multiple camera feeds provide input data capturing the indoor environment from different viewpoints.
- Data Preprocessing: Initial processing steps such as frame rate adjustment and background subtraction are performed to prepare the input data for feature extraction.
- Feature Extraction: Shape and motion features are extracted from the preprocessed video frames, capturing relevant information about human postures and movements.
- Feature Fusion: The extracted shape features (SHI) and motion features (DOF) are fused into a unified feature representation, combining both the spatial and temporal information.
- CNN-LSTM: The fused features are input to a hybrid CNN-LSTM architecture, where CNN layers extract spatial features, and LSTM layers model temporal dependencies across frames.

- Fall Detection: The learned features are used for fall event detection, where thresholding and event recognition techniques are applied to identify fall events within the video sequences.
- Classification Output: The system outputs the results of fall event detection, indicating the presence or absence of fall events in the monitored environment.

In this system, the sequential flow of data and processing steps in the fall detection system: In the first step, the fall detection system utilizes multiple camera feeds to capture the indoor environment from diverse viewpoints. These camera feeds serve as the primary input data for the system, providing comprehensive coverage of the monitored area. Before further processing, initial preprocessing steps are conducted to ensure the input data is suitable for feature extraction. This includes adjustments to the frame rate of the video streams to optimize computational efficiency and standard background subtraction techniques to segment foreground objects from the static background.

In the second step the following preprocessing, the system extracts shape and motion features from the preprocessed video frames. Shape features are derived from human silhouette images obtained through background subtraction, while motion features are computed using dense optical flow techniques applied to consecutive frames. These features capture essential information regarding human postures and movements within the monitored environment, serving as discriminative cues for fall event detection.

In the third step, the extracted shape and motion features are fused into a unified feature representation using a feature fusion approach. This fusion process combines spatial and temporal information, leveraging the complementary nature of shape and motion cues to enhance the discriminative power of the feature representation. The fused features, called Silhouette History Image (SHI) and Dense Optical Flow (DOF) Image, respectively, from the input data for subsequent processing stages.

In the fourth step, the fused features are input to a hybrid CNN-LSTM architecture, designed to capture spatial and temporal dependencies within the input data effectively. The CNN component of the architecture extracts spatial features from the fused representations, leveraging convolutional layers to learn hierarchical representations of the input features. These spatial features are then fed into LSTM layers, which model temporal dynamics across consecutive frames, allowing the system to capture the sequential nature of human actions and movements.

In the fifth step, the learned features from the CNN-LSTM architecture are utilized for fall event detection within the video sequences. This involves applying thresholding and event recognition techniques to the learned representations, enabling the system to identify instances of fall events based on predefined criteria. The combination of spatial and temporal features, along with the robust architecture of the CNN-LSTM model, facilitates accurate and reliable fall detection performance.

Finally, the system outputs the results of fall event detection, indicating the presence or absence of fall events in the monitored environment. These results provide valuable insights into the safety and well-being of individuals within the indoor space, enabling timely intervention and assistance in the event of a fall.

Background subtraction is a critical preprocessing step in the fall detection system, aimed at isolating human subjects from the static background in the video feeds. This process involves several stages to accurately detect and segment the moving foreground objects, which is essential for subsequent feature extraction and analysis.

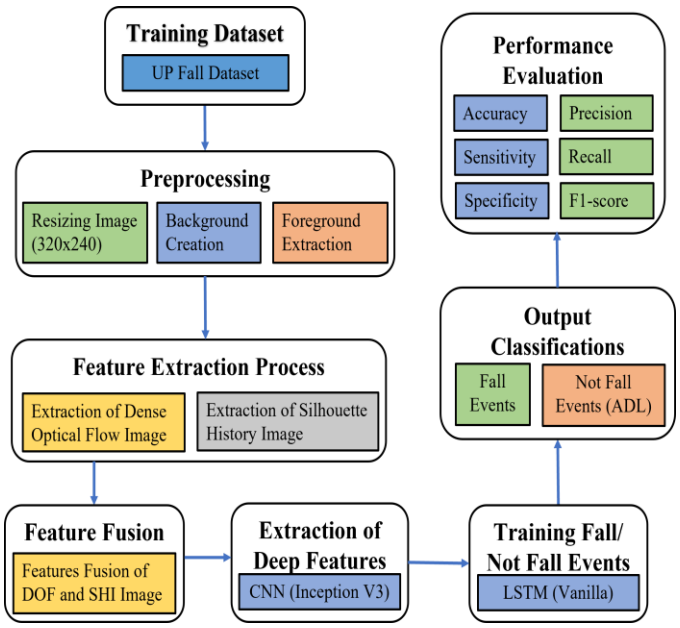


Figure 1: System flow of the advanced fall detection system using feature fusion and CNN-LSTM

3.1. Preprocessing

A) Background Creation

The first step in background subtraction is to create a background frame that represents the static elements in the scene. This is particularly challenging in fall detection scenarios where the human subject is often present throughout the video. Traditional methods like Gaussian Mixture Models (GMM) are inadequate in such cases due to their inability to handle the continuous presence of the subject. Instead, we employ a method based on frame differencing and foreground replacement:

(1) Common Background Frame (CBF) Selection: Identify a frame from the video sequence that does not contain any moving objects or humans. This frame is used as the CBF.
(2) Foreground Replacing: For videos without a clear background frame, the following steps are performed:

- Human Segmentation Mask (M): Utilize Mask-RCNN to generate a segmentation mask for the human subject.
- Pixel Replacement: Replace the pixels in the mask (M) with the corresponding pixels from the CBF using the equation:

$$BF(x,y) = \begin{cases} CBF(x,y) & if\ M(x,y) = 0 \\ F(x,y) & if\ M(x,y) = 1 \end{cases} \quad (1)$$

- Background Frame (BF) Storage: Save the resulting frame as the background frame for the video sequence.

B) Foreground Extraction

Once the background frame (BF) is established, the next step is to extract the foreground objects. This involves comparing each frame (F) of the video to the background frame to identify moving objects:

$$FG(x,y) = \begin{cases} 1 & if\ \ BF(x,y) - F(x,y) \geq TH \\ 0 & otherwise \end{cases} \quad (2)$$

The threshold (*TH*) is the pixel value that can differentiate the moving foreground and background objects. The illustration of the process of foreground extraction results is shown in Figure. 2.

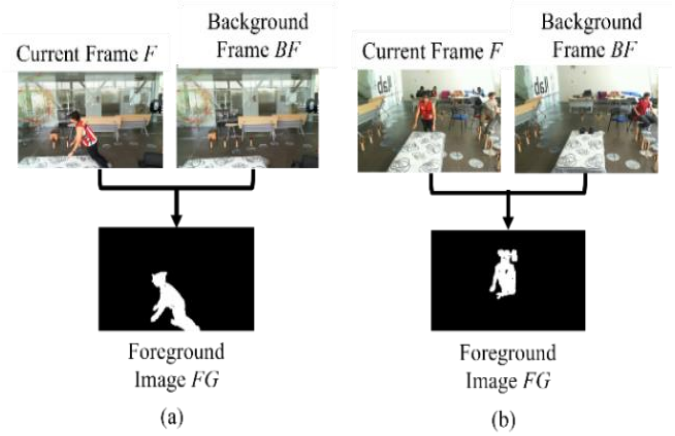


Figure 2: Illustration of foreground extraction results from (a) camera1 and (b) camera2

C) *Noise Removing*

After extracting the foreground, it is essential to filter out noise and ensure only the relevant human subjects are retained:

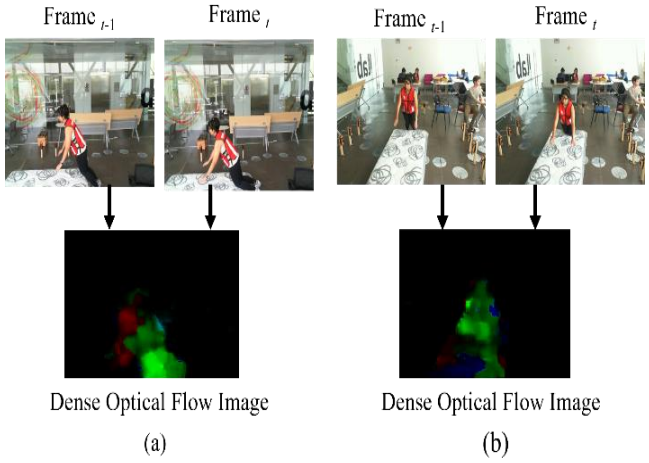


Figure 3: Background subtraction results (1st row: input frames, 2nd row: foreground

(1) Object Classification: Analyze the foreground mask to identify the human subject acting. Non-human objects are considered noise.

(2) Noise Filtering: Apply size-based filtering and morphological operations to remove small, irrelevant objects from the foreground mask. This step ensures that only the significant moving objects (humans) are retained for further processing. Some more sample images of background subtraction results are shown in Figure. 3.

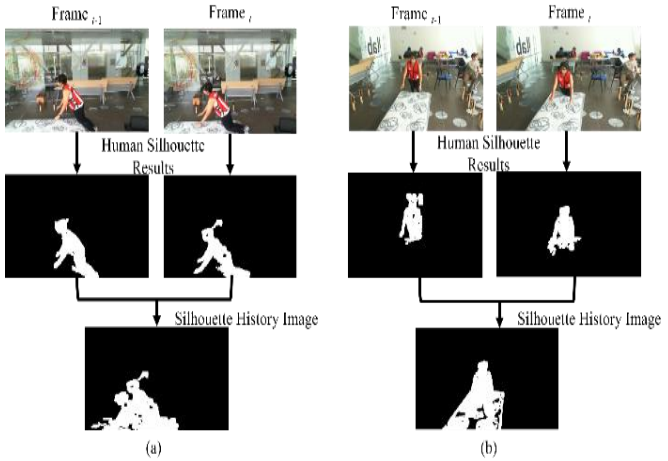


Figure 4: Creation of silhouette history image (SHI) (a) camera1 (b) camera2

3.2. *Feature Extraction*

A) *Extraction of Shape Feature*

To extract the shape feature, the edge smoothing process is performed over the noise-removed human silhouette image (foreground results). Then the resulting human silhouette images of two consecutive frames are combined to create the Silhouette History Image (SHI) results, which are used as the shape features, as shown in Figure. 4.

B) *Extraction of Motion Feature*

Dense optical flow calculation [14] is used for motion feature extraction. Dense optical flow features are extracted from every two consecutive frames. Colors are then assigned to the dense optical flow results using the HSV color space. The orientation value calculated from the dense optical flow is assigned as the Hue value, the Saturation is set to the maximum of 255, and the magnitude value of the dense optical flow is assigned as the Value in the HSV color space. The results of motion feature extraction from Camera1 and Camera2 are shown in Figure. 5 (a) and (b).

3.3. *Feature Fusion*

In this part, SHI and DOF are fused into a single input data for the training model. SHI and DOF have the same image size, and feature fusion (*FF*) is performed using the following equation. The fused feature dimensions will be the same as those of the original input images with 320×240 image size, and the result of feature fusion is shown in Figure. 6.

$$FF(x,y) = \begin{cases} SHI(x,y) & if\ SHI(x,y) = 1\ , DOF(x,y) = 0 \\ DOF(x,y) & otherwise \end{cases} \quad (3)$$

3.4. *Train CNN-LSTM for Fall Detection*

A) *Extraction of Deep Features using Convolutional Neural Network (CNN)*

A convolutional neural network (CNN) is an artificial neural

network designed to process image data and learn to classify and segment various objects within images and videos. The Inception V3 model, known for its effectiveness in image analysis and object detection, is utilized in the system to extract deep features from the input image fusion data. Inception V3, a third edition of Google's Inception CNN, consists of 42 layers. The output from the average pooling layer, a 2048-dimensional feature vector, is used as the deep features for fall detection.

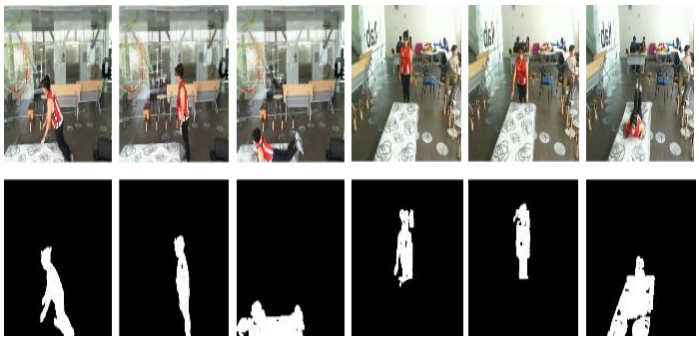


Figure 5: Motion feature extraction results (a) camera1 (b) camera2

B) Training Fall Event Detection Model using Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is designed for learning from sequential or time-series data, where the output depends on prior elements in the sequence. In this system, Long Short-Term Memory (LSTM), which consists of a cell, an input gate, an output gate, and a forget gate, is used for detecting fall events.

As shown in Figure. 7, the fused feature outputs from two cameras are fed into the Inception V3 model, pre-trained on the large ImageNet dataset. The "avg-pool" layer of Inception V3 produces a deep feature vector of length 2048. Deep features from both cameras are combined to create a feature vector of length 4096. This feature vector sequence, comprising 18 frames (spanning 3 seconds), is then fed into an LSTM for training to detect whether the input sequences contain a fall event. The LSTM used for fall detection consists of 2 stacked layers with 512 hidden units, as shown in Figure. 8. We used the ReLU activation function in two hidden layers and in the final output layer, softmax is applied for classifying the fall and not-fall events.

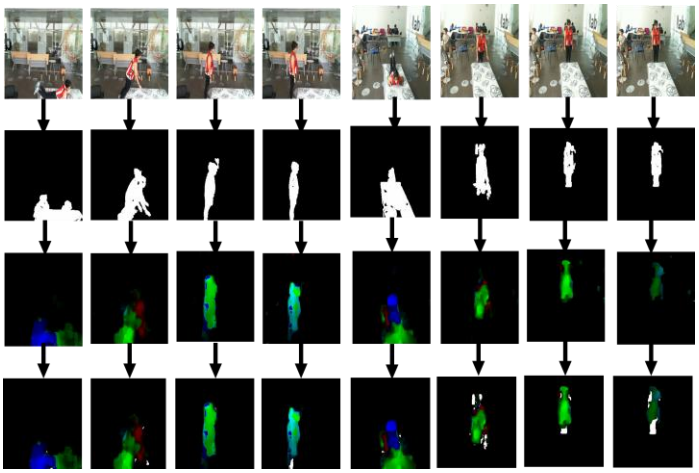


Figure 6: Sample results of feature fusion (1st row: input images, 2nd row: shape feature results, 3rd row: motion feature results, 4th row: feature fusion results)

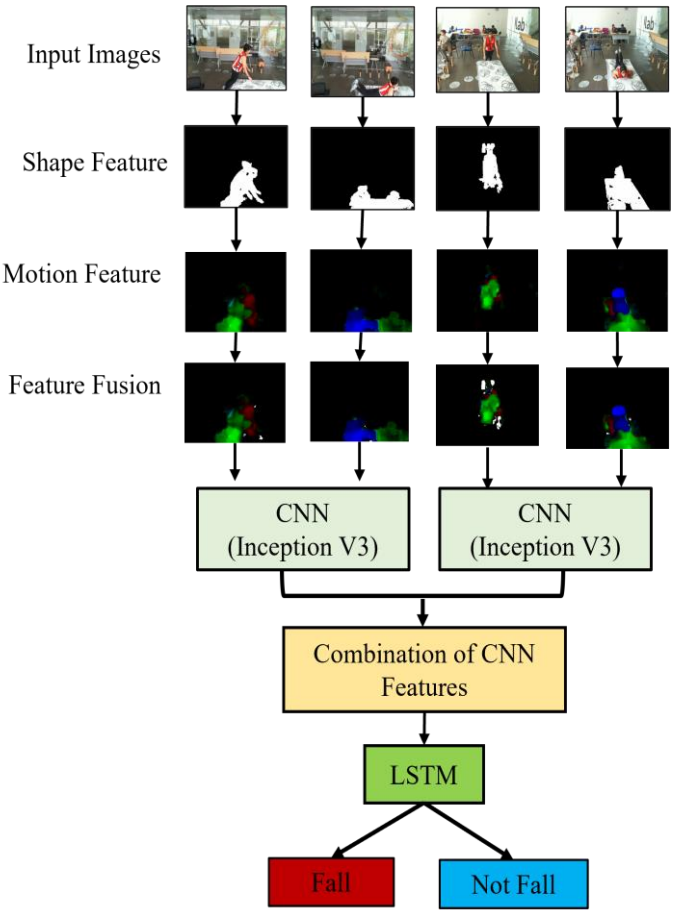


Figure 7: Flow chart of fall detection using CNN-LSTM

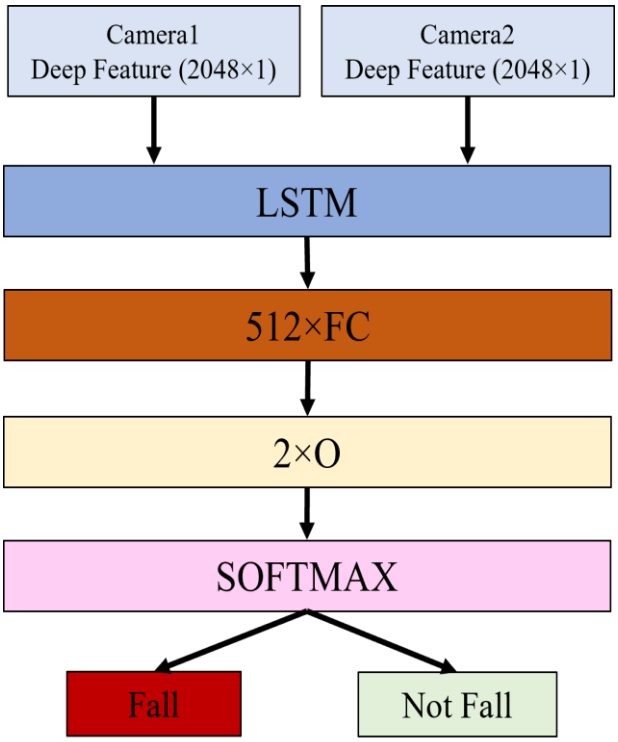


Figure 8: Architecture of fall detection model using CNN-LSTM

4. Experimental Results

4.1. Dataset

The UP-Fall Detection dataset [15], provided by Universidad Panamericana, Mexico in April 2019, includes data from 6 infrared sensors, 6 accelerometers, 3 Raspberry Pi devices, 2 cameras, and 1 brain sensor to create a multimodal dataset for fall detection. This research uses only data from the 2 cameras to implement vision-based fall detection. The dataset contains 1122 videos, each ranging from 10 to 60 seconds in length. These videos comprise 11 activities performed by 17 subjects, each repeated 3 times. Activities 1 to 5 are falls, while the remaining activities are daily living, as detailed in Table I.

The UP-Fall detection dataset provides the action videos with a frame rate of 18 fps. We use the frame rate of 6fps because most fall events take around 2 or 3 secs and according to experiments, 6fps is enough to perform the fall detection. We convert the frame rate of 18 fps into 6 fps by taking every 3rd frame from the image sequence. Then, foreground extraction is applied to 2 cameras, 3 trials, and activity 1 to 11 of all 17 subjects. The resolution of the RGB image is 320×240 and the following are some results of foreground extraction. The experiments are performed on a 2.2GHz Intel Core i7 CPU machine. The features extraction time of SHI and DOF are 0.011 s and 0.031 s respectively. The features fusion and fall detection time (3s video frames) are 0.016 s and 1.5 s respectively using Python. Some test images of the results of falls and others are shown in Figure 9.

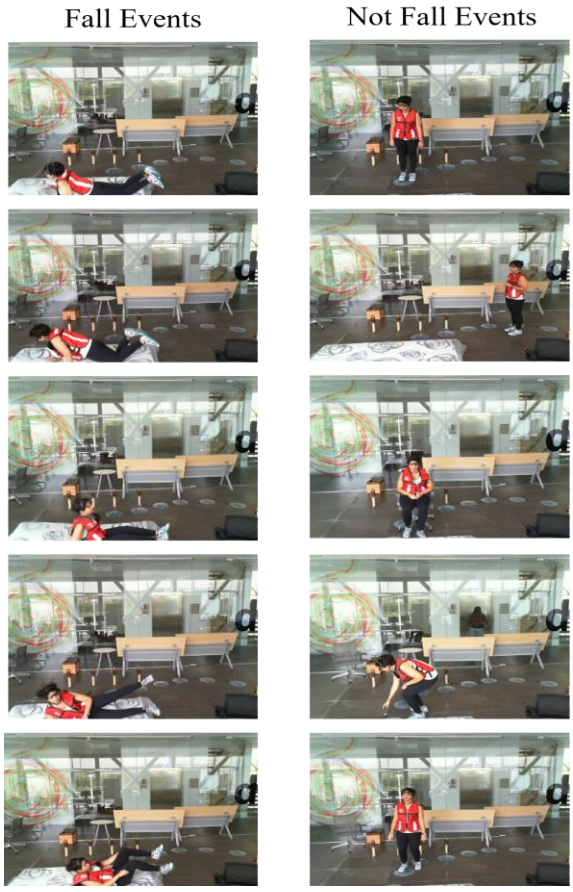


Figure 9: Some test image results of the UP-Fall detection dataset

Table 1: Activities and Their Duration

No.	Activity	Duration (sec)
1	Falling forward using hands	10
2	Falling forward using knees	10
3	Falling backward	10
4	Falling sideward	10
5	Falling while attempting to sit in an empty chair	10
6	Walking	60
7	Standing	60
8	Sitting	60
9	Picking up an object	10
10	Jumping	30
11	Laying	60

4.2. Participants

In the implementation of the advanced fall detection system, we utilized the UP-Fall Detection Dataset [16], which includes 11 activities and three trials per activity. Data were collected from over 17 participants, who were called subjects. Participants performed six simple human daily activities as well as five different types of human falls. During data collection, 17 subjects (9 male and 8 female) ranging from 18–24 years old, mean height of 1.66 m and a mean weight of 66.8 kg, were invited to perform 11 different activities for creating a comprehensive dataset for training and testing the fall detection system. Each participant's data was recorded using multiple modalities, but for this study, we focused solely on the video data captured by two cameras.

- Number of Participants: 17
- Activities: 11 distinct activities (5 fall and 6 daily activities)
- Trials: Each participant performed each activity three times, resulting in multiple video sequences for each activity.

In this research, we train 3 classification models. The first model (CNN-LSTM-2-classes) can classify only two classes such as fall and not-fall events. The second model (CNN-LSTM-7-classes) trained to classify 7 classes: fall events and other activities such as walking, standing, sitting, picking up an object, jumping, and laying. The third model (CNN-LSTM-11-classes) can classify all 11 activities as described in Table. 1.

4.3. Performance Evaluation

For fall detection performance evaluation, we trained and tested the data from the UP-Fall dataset using the same criteria as described in [9]. Data from trials 1 and 2 for 17 subjects were used as the training data, while data from trial 3 were used as the test data. To evaluate the performance of this work, the system uses the following six metrics: Accuracy, Sensitivity, Specificity, Precision, Recall, and F1-score, as given by (4)-(9),[17]. The performance evaluation of the three classification models is described in Table 2.

Moreover, we compare the performance of the proposed system with other approaches as shown in Table 3. We obtained the results of the method in [9] from their paper and used the same evaluation method to compare the results. In Table 3, we can see

that our proposed method produces higher accuracy than the method described in [9].

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Numbers\ of\ Predictions} \quad (4)$$

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \quad (6)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (7)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (8)$$

$$F1Score = 2 \times \frac{Precision.Sensitivity}{Precision + Sensitivity} \quad (9)$$

5. Discussion and Limitations

The proposed system has some limitations in the computational complexity of training the CNN-LSTM model. It needs to extract deep features using CNN and perform sequence classification using LSTM. But that limitation can be overcome by applying high-performance computing devices such as GPU-machines. Another limitation is the occlusion problem. This applied two cameras for detecting fall events. But sometimes falls can occur in an area which only two cameras cannot cover. Therefore, in the future, we plan to extend this research by applying more cameras and configuring the camera set to cover all areas of the home environment of living alone elderly.

Table 2: Performance Evaluation of Three CNN-LSTM Models (Cam1 &Cam2) on UP-Fall Detection Dataset

Models	CNN-LSTM-2 Classes	CNN-LSTM-7 Classes	CNN-LSTM-11 Classes
Accuracy (%)	99	96	93
Sensitivity (%)	98	94	79
Specificity (%)	98	99	99
Precision (%)	99	94	81
Recall (%)	98	94	79
F1-Score (%)	98	94	80

Table 3: Comparison of Fall Detection Model (CNN-LSTM-2 Classes) performance evaluation on UP-Fall Detection Dataset

Method	Espinosa R, et al [9] (Cam1 &Cam2)	Proposed (Cam1 &Cam2)	Proposed (Cam1)	Proposed (Cam2)
Accuracy (%)	95.64	99	99	99
Sensitivity (%)	97.95	98	96	98
Specificity (%)	83.08	98	96	98
Precision (%)	96.91	99	99	97
Recall (%)	-	98	96	98
F1-Score (%)	97.43	98	97	97

6. Conclusion and Future Works

In this research, a vision-based fall detection system using multiple cameras applying CNN-LSTM has been proposed. The main contribution will be taken on the "features extraction and features fusion from multiple cameras", and the architecture of CNN-LSTM for improving fall detection rate. Based on the experimental results performed on the public dataset of the UP-Fall detection dataset, the proposed system got superior performance over the state-of-the-art methods. This fact points out that the feature fusion approach for CNN-LSTM is very effective and promising for the accurate fall detection system. Limitations such as the computation complexity for training CNN-LSTM can be overcome by using high-performance computing devices. Moreover, the multi-camera approach is more cost-effective than the other multi-sensor approaches, and this research will come as applied science research which can give a lot of benefits to human society. In this research, the experiments are only performed on the UP-Fall detection, a large dataset containing 1122 action videos performed by 17 persons. Then, the proposed method got good performance results on that dataset. In the future, to confirm the effectiveness of this proposed method, we will perform more experiments on other datasets of fall detection.

Conflict of Interest

The authors declare no conflict of interest.

Author Contribution

The major portion of the work presented in this paper was carried out by the first author, Win Pa Pa San, under the supervision of the second author, Myo Khaing. Win Pa Pa San also performed the data analysis, implementation, validation, and preparation of the manuscript.

Acknowledgment

I want to extend special thanks to Dr. Myo Khaing, Professor of the Faculty of Computer Science at the University of Computer Studies, Mandalay (UCSM), and Dr. Sai Maung Maung Zaw, Professor and head of the Faculty of Computer Systems and Technologies at the University of Computer Studies, Mandalay (UCSM), for their continuous guidance, support, and suggestions.

References

[1] Q. Li, J.A. Stankovic, M.A. Hanson, A.T. Barth, J. Lach, G. Zhou, 'Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information', Proceedings - 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009, (June), 138–143, 2009, doi:10.1109/BSN.2009.46.

[2] Y. Li, K.C. Ho, M. Popescu, 'A microphone array system for automatic fall detection', IEEE Transactions on Biomedical Engineering, **59**(5), 1291–1301, 2012, doi:10.1109/TBME.2012.2186449.

[3] Y. Li, Z. Zeng, M. Popescu, K.C. Ho, 'Acoustic fall detection using a circular microphone array', 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10, 2242–2245,

2010, doi:10.1109/IEMBS.2010.5627368.

[4] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, S. Li, 'RT-Fall: A Real-Time and Contactless Fall Detection System with Commodity WiFi Devices', IEEE Transactions on Mobile Computing, **16**(2), 511–526, 2017, doi:10.1109/TMC.2016.2557795.

[5] F. Bianchi, S.J. Redmond, M.R. Narayanan, S. Cerutti, N.H. Lovell, 'Barometric pressure and triaxial accelerometry-based falls event detection', IEEE Transactions on Neural Systems and Rehabilitation Engineering, **18**(6), 619–627, 2010, doi:10.1109/TNSRE.2010.2070807.

[6] R.K. Shen, C.Y. Yang, V.R.L. Shen, W.C. Chen, 'A Novel Fall Prediction System on Smartphones', IEEE Sensors Journal, **17**(6), 1865–1871, 2017, doi:10.1109/JSEN.2016.2598524.

[7] B. Wójtowicz, A. Dobrowolski, K. Tomczykiewicz, 'Fall detector using discrete wavelet decomposition and SVM classifier', Metrology and Measurement Systems, **22**(2), 303–314, 2015, doi:10.1515/mms-2015-0026.

[8] H.U. Openpose, 'Fall Detection Based on Key Points of', Symmetry, 2020.

[9] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva, E. Moya-Albor, 'A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-Fall detection dataset', Computers in Biology and Medicine, **115**, 2019, doi:10.1016/j.compbiomed.2019.103520.

[10] S. Sherin, P.M.T. Student, A.J. Assistant, 'Human Fall Detection using Convolutional Neural Network', International Journal of Engineering Research & Technology, **8**(6), 1368–1372, 2019.

[11] A. Núñez-Marcos, G. Azkune, I. Arganda-Carreras, 'Vision-based fall detection with convolutional neural networks', Wireless Communications and Mobile Computing, **2017**, 2017, doi:10.1155/2017/9474806.

[12] K. Wang, G. Cao, D. Meng, W. Chen, W. Cao, 'Automatic fall detection of human in video using combination of features', Proceedings - 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016, 1228–1233, 2017, doi:10.1109/BIBM.2016.7822694.

[13] S. Maldonado-Bascón, C. Iglesias-Iglesias, P. Martín-Martín, S. Lafuente-Arroyo, 'Fallen people detection capabilities using assistive robot', Electronics (Switzerland), **8**(9), 2019, doi:10.3390/electronics8090915.

[14] T. Hassner, C. Liu, 'Dense image correspondences for computer vision', Dense Image Correspondences for Computer Vision, 1–295, 2015, doi:10.1007/978-3-319-23048-1.

[15] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, C. Peñafort-Asturiano, 'Up-fall detection dataset: A multimodal approach', Sensors (Switzerland), **19**(9), 2019, doi:10.3390/s19091988.

[16] L. Martinez-Villasenor, H. Ponce, K. Perez-Daniel, 'Deep learning for multimodal fall detection', Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, **2019-Octob**, 3422–3429, 2019, doi:10.1109/SMC.2019.8914429.

[17] M. Sokolova, G. Lapalme, 'A systematic analysis of performance measures for classification tasks', Information Processing and Management, **45**(4), 427–437, 2009, doi:10.1016/j.ipm.2009.03.002.

Advances in Science, Technology and Engineering Systems Journal Vol. 9, No. 6, 21-29 (2024)

www.astesj.com

Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ

ISSN: 2415-6698

Development and Application of Value *Karuta* to Understand Value in Lean Management: Initial Small-group Trial in Japan and the UK

Tamao Kobayashi, Koichi Murata*

Department of Industrial Engineering and Management, Graduate School of Industrial Technology, Nihon University, Izumi 1-2-1, Narashino City, Chiba, 2758575, Japan

ARTICLE INFO

Article history:
Received: 07 October, 2024
Revised: 05 December, 2024
Accepted: 06 Deember, 2024
Online: 12 December, 2024

Keywords:
Lean management
Value
Card Game
International Comparison
Japan

ABSTRACT

This study proposes the Value Karuta (VK), an application of the traditional Japanese card game karuta. Its goal is to contribute to the understanding of value, which is the first principle of lean management. After stating the problems of lean management and the specifications of VK, this paper confirms the validity of the proposal by discussing two surveys. The first survey explored the utility factors of the cards themselves; it was conducted with a group of students and businesspeople in Japan. The second survey observed the actual situation in the game and was conducted in a group of academics at two UK universities. Both surveys used qualitative methods, such as observation and discussion, and quantitative questionnaires. The results confirm the role of VK as a fundamental tool addressing the need to understand customer value in lean management.

1. Introduction

First, this paper is an extension of work [1] originally presented at the 2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) and contains two surveys. The original work focused on the first survey. This paper also considers the second survey for a comprehensive discussion of the proposal of Value *Karuta* (VK). This necessitated extensive revisions to the Introduction and Literature Review sections of the original work. First, we review the main text.

The difference between the Toyota Production System (TPS) and lean management is that value is added explicitly. This is the first of the five principles of lean management; the other four principles—value stream, flow, pull, and perfection—are included in the TPS. Additionally, across its legacy, there have been many cases in which the methodology has been applied to realize the four principles.

Value can only be defined by the ultimate customer [2], although its identification is difficult. This is because value designers are not customers themselves, and customers are someone else; thus, understanding the value of a customer that one has never met is difficult.

*Corresponding Author Koichi Murata, murata.kouichi30@nihon-u.ac.jp

The second principle is that of the value stream. Although its name includes the word "value," the principle focuses on waste in the value stream. In other words, value is not a direct concern. Value-stream mapping (VSM) is used as a methodology for this principle. It visualizes the overall waste in a value stream. The third to fifth principles include ideas for drastic changes to the value stream and its continuous activities. In other words, the four principles are consistent with the TPS philosophy, namely the absolute elimination of waste [3]. Most studies on lean management have focused on these four principles. The first principle has been addressed in several previous studies. However, if the customer's true value is not understood, the effects of the other principles will not be realized.

This study contributes to understanding customer value, providing an approach that helps people understand many different types of value around the world. Knowing the breadth of the value world suggests the need to understand customer value. To realize this approach, we developed VK by adapting the traditional Japanese card game *karuta*. This paper reports a survey of the initial applications of the game. Two surveys were conducted: the first was conducted in Japan with two groups of university students and businesspeople, and the second was conducted with academics at two UK universities. The first survey focused on the evaluation of the cards, which were game materials. Multiple cards were

https://dx.doi.org/10.25046/aj090603

developed, each containing an academic definition or an example of one value. The survey explored the factors that made these cards effective. The methods used included observing the actual game experience and distributing a questionnaire to the participants. The second survey confirmed whether values were truly being learned from VK, building on the perspectives of the first survey. It also evaluated whether this Japanese card game can be used internationally. These methods are the same as those used in the first survey.

The paper is organized as follows. The next section describes a way of thinking about value in lean management, conventional tools for understanding value, and the relationship between conventional and developed tools. Section 3 describes the research procedure based on two surveys in Japan and the UK in which the game was implemented and evaluated using a questionnaire. Section 4 presents and discusses the research results. Finally, section 5 presents the conclusions of the study.

2. Conventional and Developed Tools

2.1. Value

While searching the literature on values, we discovered that many types of value have been studied. This section reviews 24 types of values by 15 authors of works spanning approximately 150 years, from Marx's time to recent years. These types of value are used in the VK and can be classified into the following four academic fields:

Economics literature illustrates seven types of value: use, exchange, perceived, acquisition, transaction, firm, and intangible value [4-7]. The first four value types were studied relatively early in this review. The fifth and sixth value types were considered in recent years. In seventh value type, intangible resources are becoming what gives firms competitive power.

Psychology literature illustrates two types of value: terminal and instrumental [8]. These two value types comprise the Rokeach Value Survey. Each value has 18 subcategories that organize the essential attributes of human beings. This system continues to be used in numerous investigations.

Sociology literature illustrates three types of value, linking value, cultural value, and dominant social value [9]-[11]. Value in this field expresses social phenomena in the relationships between people. For example, the dominant social value states that the K-pop boom in Korea is a contemporary social phenomenon [11].

Marketing literature illustrates 12 types of value: experience, basis, convenience, sensory, idea, customer, context, consumption, semantic, sticking, self-expression, and environmental value [12]-]19]. Marketing includes the most value types of the four academic fields reviewed in this study. It flourished around the year 2000 and interpreted how people attached meaning to products. Multiple value types are a characteristic of works authored in this field. For example, Wada [14] and Nobeoka [18] propose four and three values, respectively.

This review shows that value has been studied in many fields, especially marketing, which is closely related to the customer. However, no such research has been conducted in the field of lean management.

2.2. Value in Lean Management

How, then, is value handled in lean management? Value is the first of the five principles of lean management [3]. When one reads into the related literature, one learns that if value is not accurately defined, it will be skewed by value chain functions such as strategy, engineering, supplier, and sales. The skew of value (SoV) within each function is as follows [3]:

SoV 1: Preexisting organization-oriented

"Business school-trained senior executives of American firms tell us about their short-term competitive problems and the consequent cost-cutting initiatives."

SoV 2: Technology-oriented

"Designs with more complexity produced with ever more complex machinery were asserted to be just what the customer wanted and just what the production needed."

SoV 3: Supplier relationship-oriented

"The immediate needs of employees and suppliers were prioritized over the needs of the customer, which must sustain any firm in the long term."

SoV 4: Preexisting service-oriented

"Many producers want to make what they are already making. And then, many customers only know how to ask for some variant of what they are already getting."

Lean management has proposed dialogue to overcome SoV. Many value chain players only imagine customer value. If value is something they have never seen before, they will never be able to reach it even if they have a dialogue with the customer.

2.3. Tools to Understand Value in Lean Management and Others

Tools to understand value have been developed outside lean management. The following reviews value proposition (VP) and value engineering (VE) in addition to VSM in lean management.

VSM is a well-known method for identifying waste and improving performance proposed in the lean-manufacturing approach [20], [21]. This tool has been used for process improvement [22] and has been illustrated graphically [23]. Its primary goals are process modeling, investigating process waste, estimating the lead time associated with a certain product flow throughout a system, and estimating process efficiency [24].

The VP is a multifaceted bundle of products, services, prices, communication, and interactions that customers experience in their relationship with the supplier [25]. This conceptualization of the VP as a multifaceted bundle enables a better understanding of the complexity that emerges when integrating sustainability into the value propositions of business models [26].

VE refers to processes designed to reduce costs while maintaining standards [27]. VE complements the target cost and increases the chances of simultaneously reaching cost targets and guaranteeing quality [28].

Although these tools contain the word 'value' within their names, they only consider factors other than value.

2.4. Proposed Value Tool

VK is a valuable tool developed with the concept of "learning value in a fun and easy-to-understand manner" in mind. We have developed VK to be enjoyed as a game. Japan has several indoor games, and *karuta* is a traditional Japanese playing card game [29], [30]. VK was created with reference to the layout and rules of *karuta* [31].

Karuta contributes many functions to Japanese culture, such as community creation and spiritual fulfillment in daily life. It is played in classrooms and family gatherings in Japan, whereas European card games often feature in gambling [32]. *Karuta* also maintains a religious record and features as decoration of Buddhist shrines [33]. Today, competitive *karuta* is a popular sport. Players analyze how to improve their skills using the latest motion capture technology [34]. In an era of low birth rates and an aging society, *karuta* contributes to intergenerational social interactions between older people and children in Japan [35]. Furthermore, it has spread to other Asian countries. For example, an elementary school in Indonesia tried to use *karuta* as a tool for language education [36]. This highlights how *karuta* can have a knowledge acquisition function.

Value *karuta* is designed to understand the 24 types of values described in Section 2.1.

Two cards are used for each value. One was *torifuda* and the other was *yomifuda*. The *torifuda* consists of a front side with the name of the value and an illustration of the corresponding value, and a back side with the name, concept, and outline of the value. The *yomifuda* has three features: the name of the value, an overview, and an easy-to-understand explanation. In the game, one person reads the *yomifuda* (reader) and multiple people read the *torifuda* (takers). The rules are as follows: (1) all cards in *torifuda* are arranged in front of the takers; (2) the reader then reads one *yomifuda*; (3) the takers compete to pick up the *torifuda* with the name of the value read by the reader; and (4) steps (2) and (3) are repeated until no cards remain. Victory or defeat in the game is determined by the number of *karuta* cards taken. Both readers and takers can learn about value through the game. Figures 1 and 2 show examples of *karuta* cards. The advantage of this value tool is that it can be played like a game, and the value can be understood from multiple sources of information. Players receive the visual information of the characters and illustrations written on the *karuta*, and they receive auditory information read during the *karuta* game. Additionally, because *karuta* is a traditional Japanese game, the tool is easily accepted by the Japanese people. A disadvantage of the tool is the need to explain the game to people unfamiliar with *karuta*, including foreigners. Another disadvantage is the fact that the game can only be played with multiple people and not as a single-player game.

3. Research Method

This study consisted of two surveys, as shown in Figure 3. The first survey aimed to evaluate VK materials. The materials refer to the cards that play a central role in the game. The game's success or failure depends on the quality of the materials, and this motivated the first survey. The second survey evaluated the game experience. It aimed to clarify the players' feelings toward the game their impressions of it, and what they learned from the game.

Figure 1: Value Karuta (Japanese version) (From left to right, torifuda, torifuda back, yomifuda)

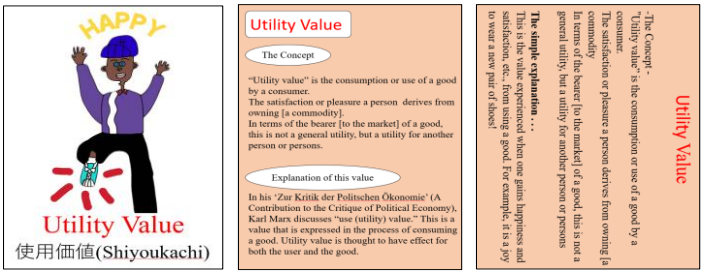


Figure 2: Value Karuta (English version) (From left to right, torifuda, torifuda back, yomifuda)

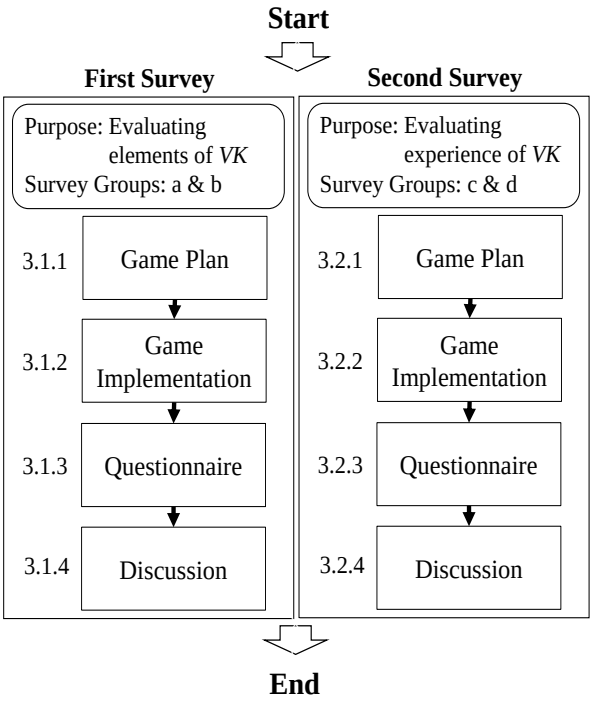


Figure 3: Survey Procedure

Both surveys followed the same four steps: game plan, game implementation, questionnaire, and discussion. Details of the procedure for each survey are provided below.

3.1. First Survey

3.1.1 Game Plan

The survey involved two groups, Group a and Group b. Group a consisted of 13 college students in their twenties, six men and seven women. Two male students were foreigners (Chinese and Turkish), whereas the other students were Japanese.

Group b consisted of seven practitioners of different ages: one in their twenties, two in their thirties, two in their forties, and two in their fifties. There were four males and three females. All were Japanese. Their industries included manufacturing, retail, tax accounting, and real estate.

3.1.2 Game Implementation

Each group played the game twice. When conducting the experiment, we prepared an environment in which the cards were spread out, and the participants could sit around them.

The game for Group a was held at a university seminar camp. In the gymnasium, cards were laid out on the floor, and the game was played while players sat on the floor. The game for Group b was held as an icebreaker for a seminar on lean management. The cards were arranged on a table in the seminar venue, and the participants sat on chairs to play the games.

3.1.3 Questionnaire

After the game ended, we distributed a questionnaire consisting of Questions A–G. Question A was, "Were you satisfied?" This was the players' overall rating of the game. The respondents were asked to respond on a three-point scale (satisfied, neutral, and dissatisfied) and state the reasons for their responses. Question B was, "How was the visibility of the characters on the table of the *torifuda*?" Question C was, "How was the visibility of the characters on the back of the *torifuda*?" Question E was, "How was the visibility of the picture?" These four questions were used to evaluate *karuta*. There were three levels: bad, normal, and good. Question D was, "How much did you enjoy the game?" The aim was for players to have fun. The respondents answered on a six-level scale: 100%, 80%, 60%, 40%, 20%, and 0%. Question F was, "Please write down your favorite value." This item investigated which value players tend to recall and therefore which value tends to leave an impression. Last, Question G was, "Please write down any good points or concerns you have about this game in free description."

3.1.4 Discussion

We analyzed whether the game was effective in promoting the understanding of value. From the results of the questionnaire, we extracted the internal and external factors that promoted the understanding of value. Issues raised by these results were examined further in the second survey.

3.2. Second Survey

3.2.1 Game Plan

The survey had two groups, Group c and Group d. Group c consisted of 15 academics from different UK universities: three in their twenties, seven in their thirties, one in their forties, one in their fifties, and three of unknown ages. There were eight males, four females, and three of unknown gender. Nationality was mixed.

Group d consisted of six academics from a UK university not included in Group c. Three were in their thirties, one in their forties, one in their fifties, and one of unknown age. Nationality was mixed.

3.2.2 Game Implementation

In the first survey, the game was played as part of an event. Before the game, the participants received an explanation of the game rules. In the second survey, the game was played after receiving a detailed explanation of its background and purpose.

Both groups' games were held at UK universities. Another purpose of this second survey was to understand whether *karuta*, a card game that is culturally Japanese, would be accepted in an international environment. For this purpose, we also created and used *karuta* translated from Japanese into English. The cards were made of Washi paper, which is a traditional Japanese paper.

Both groups played the game twice. They arranged cards on a table in the meeting room and sat in chairs to play the game.

3.2.3 Questionnaire

The questionnaire for the second survey consisted of Questions A2–G2. Question A2 was, "Which types of value left an impression on you?" The participants were asked to write an answer regarding the type of value that made an impression on them while playing the game. Question B2 was "How well do you understand value?" Using a five-point scale (5 = understand well, 1 = do not understand), we evaluated whether participants were able to understand value by playing the game. Question C2 was, "Please tell us why you chose the number in Question B2." This item was used to determine the points at which the participants understood the value during the game. Question D2 was, "Please tell us about the difficulty level of VK." Using a five-point scale (5 = easy, 1 = difficult), participants were asked to rate how easy it was to play the game. Question E2 was, "Please tell us your impression of the appearance of VK." It asked the participants how they felt about the game's appearance. The evaluation was performed by selecting one or more of the following seven items: pretty, pleasant, bright, cool, sober, quiet, and other. Question F2 was, "Please tell us about a scene in the game that left an impression on you." This aimed to determine when players concentrate while playing the game. The participants were asked to select one or more of the following four items: "When taking a card," "When looking for a card," "When deciding the winner," and "Other." Question G2 was, "If you have any other comments or opinions, please let us know." These questions sought opinions on aspects of the game other than those addressed in the previous six questions.

3.2.4 Discussion

This step demonstrates the possibility of promoting an understanding of value through VK. The first survey confirmed the effectiveness of the game's materials, specifically the cards themselves. Additionally, the second survey confirmed whether VK would be enjoyable for players who were unfamiliar with the game and whether these players had time to think about value.

4. Research Results

4.1. First Survey Results

Table 1 shows the questionnaire answers by Group a. Twelve people answered "Satisfied," and one person answered "Neither" to Question A. Men who answered "Satisfied" were satisfied with the overall design of the game and *karuta*. The women were satisfied with the cuteness of the illustrations and the rules of the game. Men who answered "Neither" were dissatisfied with the

game's outcome. For Question B, 12 people answered "Good," and one person answered "Bad." For Question C, 12 people answered "Good," and one person answered "Normal." For Question D, 12 out of 13 people answered "100%," and one answered "80%." For Question E, 11 out of 13 people answered "Good," and two answered "Bad." For Question F, two people listed the link and commitment values (respectively), and the other eight values were each listed by one person. These are the value types whose meanings can be inferred from their names. Question G was answered by 8 out of 13 people, five of whom were anonymous. Men's impressions considered how to play *karuta* and the environment. Women's impressions considered the cuteness and fun of *karuta*.

Table 1: Questionnaire Results for Group a

Question	Evaluation	Number of people	Gender		Comment
			Men	Women	
A	Satisfied	12	5	7	*1
	Neither	1	1	0	*2
	Dissatisfied	0	0	0	−
B	Bad	1	1	0	
	Normal	0	0	0	
	Good	12	5	7	
C	Bad	0	0	0	
	Normal	1	1	0	
	Good	12	5	7	
D	100%	12	5	7	
	80%	1	1	0	
	60, 40, 20, 0%	0	0	0	
E	Bad	2	1	1	
	Normal	0	0	0	
	Good	11	5	6	
F	Link Value	2	1	1	
	Sticking Value	2	2	1	
	Intellectual Value	1	1	0	
	Semantic Value	1	1	0	
	Convenience Value	1	1	0	
	Corporate Value	1	0	1	
	Environmental Value	1	0	1	
	Exchange Value	1	0	1	
	Social Value	1	0	1	
	Transaction Value	1	0	1	
G	With Comments	8	3	5	*3
	No Comments	5	3	2	

<Comments on Question A>

*1: That was very fun. / The text and images were easy to understand. The fact that an English notation was included was good. / It was good to know various types of value. / I learned a lot of different values. / I learned a lot of new values. The pictures were cute and fun. / I was able to know the value that I do not usually use. / Good to know the value. /Aiming to be number one, we were able to do it while cooperating. / It was fun. The difference in game activity was easy to understand. / It was fun.

*2: The number is slightly less.

<Comments on Question G>

*3: It is hard to judge by looking at a picture. / Good to learn about value. / Perfect with cushions. / The letters were easy to read, and the pictures were cute. / The pictures were so cute and funny. / I was able to study in an easy-to-understand manner and enjoyed it. / It was nice to have cute pictures on all the karuta cards. I had fun. Thank you. / Good to learn about value. / The writing on the back of the card was a little hard to read. However, I thought it would be easy to understand and global owing to the word notation.

Regarding Group b (Table 2), five people answered "Satisfied," and two answered "Neither" to Question A. Men who answered "Satisfied" commented on the rules of *karuta* and their impressions of *karuta* itself, as well as their nostalgia for playing *karuta*. One commented on the women he enjoyed playing with and how nice the illustrations were. Those who answered "Neither" offered advice on how to improve the rules to achieve the game's goals. All respondents answered "Good" to Questions B and C. Question D was 100% for four people and 80% for three people. All the respondents answered "Good" to Question E. Question F assessed the ability, semantics, and instrumental values. All the respondents answered Question G. Participants primarily raised suggestions for improving the *karuta* rules.

Table 2: Questionnaire Results for Group b

Question	Evaluation	Number of people	Gender		Comment
			Men	Women	
A	Satisfied	5	4	1	*1
	Neither	2	1	1	*2
	Dissatisfied	0	0	0	−
B	Bad	1	1	0	
	Normal	0	0	0	
	Good	7	4	3	
C	Bad	0	0	0	
	Normal	1	1	0	
	Good	7	4	3	
D	100%	4	2	2	
	80%	3	2	1	
	60, 40, 20, 0%	0	0	0	
E	Bad	0	0	0	
	Normal	0	0	0	
	Good	7	4	3	
F	Ability Value	1	1	0	
	Semantic Value	1	0	1	
	Instrumental Value	1	0	1	
	Perceived Value	1	1	0	
	Intangible Value	1	0	1	
	Link Value	1	1	0	
	Unanswered	1	1	0	
G	With Comments	7	5	3	*3
	No Comments	0	0	0	

<Comments on Question A>

*1: It was fun. I think I noticed a lot. However, I think it would be more interesting if the rules were stricter than the karuta itself. / I played karuta for the first time in a while. / I really liked the illustrations. I had a lot of fun learning about values. I had no idea it was worth so much. Please keep doing a good job. / I was able to learn about value while having fun. I learned a lot. / The illustrations were very nice.

*2: It was fun but difficult. / The content was very interesting and a new experience, but I still do not fully understand it.

<Comments on Question G>

*3: It was hard to tell the difference between the values. / The value and its explanation just could not connect, but it was fun. / It is the perfect icebreaker because moving your body creates conversation and stimulates your intellectual curiosity. Awareness of "heh" is a strong motivation to read the text. / It was great to get to know each other as an icebreaker, and it was a great opportunity to learn about values. But the best part was being able to talk to all the students. It was great because I do not usually talk with students. / The pictures were so unique and cute! I think I would have been more absorbed in playing *karuta* if I had had more time to deepen my understanding. It has been a long time since I have played karuta in this form, and it was a lot of fun! / I was able to learn the value through the game (*karuta*) using pictures and easy-to-understand words, so I thought it would be a good idea to get it into my head more smoothly. I would like to hear a more detailed explanation of each value. / I think you can enjoy learning about value through *karuta*. I think it was good that I chose *karuta* as a tool. How about

creating a *karuta* role with rules that are conscious of the understanding and connection of the game and value of *karuta*? Five points if you have XX value, XX value, and XX value, which are connected to XX.

The results of the questionnaire survey confirmed three factors that make the VK effective: (1) nostalgia and design, (2) Japanese naming and type, and (3) the implementation environment. Each of these factors is discussed below.

Regarding nostalgia and design, two observations can be drawn from the questionnaire results. First, *karuta* is a game played by Japanese people when they are young. This is evident from the comment by the businesspeople group: "I played *karuta* for the first time in a long time." Second, the college students expressed many opinions about the illustrations, such as "The pictures were very cute and interesting" and "I'm glad that all the *karuta* cards have cute pictures on them." The illustrations on the *karuta* cards create familiarity.

For naming and Japanese type, college students and businesspeople listed two value types in Question F: linked value and semantic value. Among the 34 values, the linked value was the only one in Japanese *katakana* notation. From the college students' answers, multiple people listed the sticking value. Among the 34 types, the commitment value was the only one in *hiragana* notation. Because the other 32 symbols were written in Chinese characters, it can be assumed that they left an impression. Additionally, it is easy to imagine the meaning of the word "stickiness," and the word "link" has an image that makes one wonder and want to investigate. Both are attractive words that modify value, suggesting that the naming and notation of types of value influence their understanding.

Third, the implementation environments differed between the two groups. The college student group held an event during a seminar camp, whereas the businesspeople held an educational seminar. Comparing the comments on Question G, the college student group had monotonous impressions of the game, such as "It was fun" and "It was interesting." The businesspeople group said, "I want to hear more detailed explanations about each value." This answer makes one aware of technical aspects, such as the rules of VK and the motivation for value learning.

4.2. Second Survey Results

Regarding Group c (Table 3), responses were received from 15 people: eight were men, four were women, and three did not answer. Their ages ranged from their twenties to their fifties, with an average of 30 years. For Question A1, three people answered "Link Value," and two answered "Cultural Value" and "Dominant Value," respectively. For Question B2, the average level of understanding of value was 3.6, and for Question D2, the average level of difficulty was 3.4. From Question E2, 20% of participants answered "pretty" and "bright" regarding the cards' appearance. From Question F2, 53% of participants answered "when looking for a card" regarding the most memorable moment. Nine participants provided a free-form descriptions in Question G2.

Six people answered in Group d (Table 4). Four patients were men and two were women. Their ages ranged from their twenties to their fifties, with an average of 30 years. For Question A2, six respondents provided different answers regarding memorable values. For Question B2, the average understanding of the value types was 3.8, and the average difficulty of Question D2 was 3.5. From Question E2, 67% answered "Pretty" regarding the appearance of the card. From Question F2, 67% answered "when looking for a card" regarding memorable moments. Everyone provided free-form descriptions for Question G2.

Table 3: Questionnaire Results for Group c

Question	Evaluation	Number of people	Gender			Comment
			Men	Women	No Response	
A2	Link Value	3	3	0	0	
	Dominant Value	2	2	0	0	
	Cultural Value	2	2	0	0	
	Environmental Value	1	1	0	0	
	Social Value	1	1	0	0	
	Transaction Value	1	0	0	0	
	Social Value	1	0	0	0	
	Firm Value	1	0	0	0	
	Terminal Value	1	0	0	0	
	Convenience Value	1	0	0	0	
	Intangible Value	1	0	0	0	
	Context Value	1	0	0	0	
	Perceived Value	1	0	0	0	
	Self-Expression Value	1	0	0	0	
	Idea Value	1	0	0	0	
B2	1 (Not Understood)	0	0	0	0	–
	2	3	1	2	0	*1
	3	3	1	2	0	*2
	4	6	4	0	0	*3
	5 (Well Understood)	3	2	0	0	*4
D2	1 (Difficult)	0	0	0	0	
	2	2	1	0	1	
	3	6	3	2	1	
	4	3	2	1	0	
	5 (Easy)	2	1	0	1	
E2	Pretty	7	6	0	1	
	Pleasant	5	3	1	1	
	Bright	7	4	2	1	
	Cool	6	3	1	2	
	Sober	2	2	0	0	
	Quiet	1	0	1	0	
	Other	2	1	0	1	*5
F2	When Taking a Card	4	1	0	3	
	When Looking for a Card	9	4	4	1	
	When Deciding the Winner	2	1	0	1	
	Other	1	1	0	0	*6
G2	With Comments	9	5	1	3	*7
	No Comments	6	3	3	0	

<Answers to Question C2>

*1: The value is a bit hard to understand. / Some values are hard to understand. / Because my English is not good. So interesting is better.

*2: New value for me. / I could not hear the reader clearly. / I am not familiar with each value.

*3: New topic to me - Enjoyed learning while playing Karuta. / I can understand more types of value. / Visual link between theory and images is practical. / It affects us. / I do not have knowledge on this topic, but the game helps.

*4: It was easy with the image and hints in the description. / I can understand most meanings with the cards.

<Other comments of Question E2>

*5: Lose! / Fun / Entertainment

<Other comment of Question F2>

*6: Listening intently.

<Comments on Question G2>

*7: I think it would be better to first let the player familiarize themselves with the intention of the concept before they play. / This is a good game. / Sometimes, it is not clear what the reader read. -> It's a fun game. / Thank you. It was a good idea to use this game. / Thank you.

Table 4: Questionnaire Results for Group d

Question	Evaluation	Number of people	Gender		Comment
			Man	Women	
A2	Customer Value	1	1	0	
	Cultural Value	1	0	1	
	Perceived Value	1	0	1	
	Consumption Value	1	1	0	
	Other	2	2	0	*1
B2	1 (Not Understood)	0	0	0	–
	2	0	0	0	
	3	2	1	1	*2
	4	3	2	1	*3
	5 (Well Understood)	1	1	0	*4
D2	1 (Difficult)	0	0	0	
	2	0	0	0	
	3	2	0	2	
	4	2	2	0	
	5 (Easy)	1	1	0	
E2	Pretty	7	1	2	
	Pleasant	5	4	1	
	Bright	1	1	0	
	Cool	3	3	0	
	Sober	0	0	0	
	Quiet	0	0	0	
	Other	1	1	0	*5
F2	When Taking a Card	2	1	1	
	When Looking for a Card	3	2	1	
	When Deciding the Winner	1	1	0	
	Other	0	0	0	
G2	With Comments	6	4	2	*6
	No Comments	0	0	0	

<Answer to Question A2>

*1: Friendship / Selecting the correct answer was not easy.

<Comments on Question C2>

*2: Previous experience with research. / Value is sub stateless.

*3: Value is a spoof topic and needs further discovery. / This game made me understand different types of value. / Because of my past research on value.

*4: Experience

<Comment on Question E2>

*5: Paper^^

<Comments on Question G2>

*6: Great Game! I had a lot of fun! Thanks! / Thank you, was cursed. / Thank you! / Very pleasant way to learn about value. / Great game to engage with. Fun game, great way to teach and clarify different concepts. Great! / It is a very nice game.

The questionnaire results showed that the game has mostly been well received in the international environment. To be clear, some comments could be interpreted as lip service.

For Question E2 on the appearance of the card itself, "Pretty" was the most common impression for both groups. This demonstrates a similar tendency to what was observed in the first survey. Players, therefore, have positive experiences of the game, regardless of nationality.

Questions D2, F2, and some comments suggested that the game was not difficult or incomprehensible, and the participants were able to concentrate on it. In particular, many participants selected "When looking for a card" in Question F2.

Descriptions of everyday life written on *torifuda* cards are easy for Japanese people to understand and are important hints for finding cards. However, some players commented that the game was too Japanese and difficult for them to understand without cultural context. It is therefore debatable whether the purpose of the game is to understand Japanese culture or types of value.

In both groups, participants found conversations about types of value worthwhile. Even if they chose the wrong card, their mistakes increased their fun as players of the game. This was rarely seen when the game was played in Japan. This difference may provide cross-cultural insights.

From the results of Question A2, as in Japan, there was little overlap in the types of value that made an impression on participants. This result shows that, even if people live in the same environment, they resonate with different value types. In other words, it demonstrates the need to sincerely perceive values other than one's own. This finding implies that the first principle of lean management is important.

4.3. Discussion of Both Surveys' Results

Lean management requires dialogue to understand customer value. Customers are entirely unknown to value chain functions. It is even difficult to understand one's own family and friends. However, value chain functions must understand their customers' needs. Moreover, this understanding includes economic considerations that skew the true value. To address this challenge, this study proposed a new card game. Two surveys were conducted to confirm that the card game is useful for generating dialogue on value. The corresponding results are summarized as follows:

First, the proposed card game utilizes the fun characteristics of card games. In addition, the card game players accepted pictures of each value drawn on each card. Furthermore, the card game is enjoyable for both Japanese people and foreigners. Japanese people can enjoy the card game as a nostalgic experience echoing childhood memories, and non-Japanese people can gain a good experience of Japanese culture. The material and cultural aspects of the card game create a shared platform for producing dialogue between card game players.

Second, each card has one meaning. However, each player has a slightly different interpretation of each card and its meaning. This situation promotes dialogue while playing games. In the original *karuta*, players compete on the number of cards they take, so taking the wrong card leads to a loss. However, in the proposed

karuta, they may experience more enjoyment sharing their understanding of a value when taking the wrong card.

5. Concluding Remarks

In this study, new tools were developed for learning about and applying value. Its utility mainly lies in the ingenuity of providing knowledge about value types using *karuta* with a game-like nature.

In general, acquiring academic knowledge is difficult. *Karuta* was used to ease this difficulty, and simple sentences and pop pictures were found to be effective. This multiplayer game generated substantial conversations among groups in the UK. This phenomenon promoted an understanding of value.

This new tool for directly thinking about value brings benefits to both industry and academia. For industry, by combining a game-like experience with indirect analysis tools such as VSM, VP, and VE, their functions are expanded. This leads to a richer analytical perspective with clearer recognition of the original purpose of VSM, VP, and VE, which is to improve value. For academia, the proposed tool combines a conceptual approach with a practical methodology for value studies. In doing so, it will contribute to promoting the application of understandings of value throughout society.

There are two possible future studies on this topic. The first aims to improve the game to further promote the understanding of value, and the second adjusts the design of this game to help businesspeople. The first strengthens the motivation to understand value. Card games, including *karuta*, have a completion principle: they promote motivation to win. The rules of the game can be improved using this principle. The second study addresses how to utilize the learnings from the game. This relates to the first principle of lean management and supports the understanding that various values exist. Customers of this tool will be predominantly businesspeople. They strive to reduce waste and increase profits. To contribute to their goals, this study must specifically define the role of the game in their activities.

A limitation of this study is that it was based on a simple questionnaire, and because the survey samples were small, the sample size should be increased for statistical analysis in future studies. A more elaborate experimental design will aim to verify earlier results. It is also hoped that more people will experience and understand VK and that the system will be developed to enrich their daily lives. This ought to contribute to the acquisition of new words and awareness of concepts that can be freely expressed in the real world. VK also focuses on understanding the types of value. Therefore, one limitation is how these values can be applied to actual problems. In future, the development of a methodology to address this challenge of application should be considered.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] T. Kobayashi, Y. Ishizaki, H. Tukamoto, M. Sugi, M. Nakane, K. Murata, "A Study on Utility Factors of Value Karuta-Application to College Student and Business Person Groups," in 2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 0865-0869, 2023, doi: 10.1109/IEEM58616.2023.10406441.

[2] J. P. Womack, D. T. Jones, Lean thinking: Banish waste and create wealth in your corporation, Revised and updated, Free Press, 2003.

[3] T. Ōno, Toyota production system: Beyond large-scale production, Productivity Press, 2019.

[4] K. Marx, trans. B. Fowkes, Capital vol. 1: A critique of political economy, Penguin Classics, 1992.

[5] K.B. Monroe, Pricing: Making profitable decisions, McGraw-Hill, 1990.

[6] P.G. Berger, E. Ofekb, "Diversification's effect on firm value," Journal of Financial Economics, **37**(1), 39-65, 1995, doi.org/10.1016/0304-405X(94)00798-6.

[7] L.A. Eisfeldt, E. Kim, D. Papanikolaou, "Intangible value," **28056**, National Bureau of Economic Research, 2020, doi:10.3386/w28056.

[8] M. Rokeach, The nature of human values, Free Press, 1973.

[9] B. Cova, "Community and consumption: Towards a definition of the "linking value" of product or services," European Journal of Marketing, **31**(3/4), 297-316, 1997, doi.org/10.1108/03090569710162380.

[10] C. Marzilli, "Concept analysis of value," International Journal of Recent Advances in Multidisciplinary Research, **3**(11), 1919-1921, 2016, hdl.handle.net/10950/1197.

[11] E. Nesmeyanov, Y. Petrova, R. Bachieva, O. Vasichkina, "The concept of value in modern youth subcultures of K-pop and brony in the period of globalization," SHS Web of Conferences, **72**, 1-6, 2019, doi.org/10.1051/shsconf/20197203025.

[12] B.H. Schmitt, Experiential marketing: How to get customers to sense, feel, think, act, relate, Free Press, 1999.

[13] B.H. Schmitt, A. Simonson, Marketing aesthetics, Prentice Hall, 1997.

[14] M. Wada, Brand Planning and Brand building (Burando kachi kyouso) (in Japanese), Dobunkan Publisher, 2002.

[15] B. Dodds, Managing customer value: Essentials of product quality, customer service, and price decisions, University Press of America, 2003.

[16] J.E. Finch, "The impact of personal consumption values and beliefs on organic food purchase behavior," Journal of Food Products Marketing, **11**(4), 63-76, 2006, doi.org/10.1300/J038v11n04_05.

[17] L.S. Vargo, R. F. Lush, "Evolving to a new dominant logic for marketing," Journal of Marketing, **68**(1), 1-17, 2004, doi.org/10.1509/jmkg.68.1.1.24036.

[18] K. Nobeoka, "Creation of premium value in new product development to avoid commoditization," Journal of Economics and Business Administration, **194**(6), 1-14, 2006, doi: 10.24546/00056119 (in Japanese).

[19] E. Fraj, E. Martinez, "Environmental values and lifestyles as determining factors of ecological consumer behavior: An empirical analysis," Journal of Consumer Marketing, **23**(3), 133-144, 2006, doi.org/10.1108/07363760610663295.

[20] S. Ghosh, K. Lever, "A lean proposal: development of value stream mapping for L'Oreal's artwork process," Business Process Management Journal, **26**(7), 1925-1947, 2020, doi.org/10.1108/BPMJ-02-2020-0075.

[21] B. Singh, S.K. Garg, S.K. Sharma, "Value stream mapping: Literature review and implications for Indian industry," Advanced Manufacturing Technology, **53**(5-8), 799–809, 2011, doi: 10.1007/s00170-010-2860-7.

[22] A. Dadashneiad, C. Valmohammadi, "Investigating the effect of value stream mapping on operational losses: A case study," Journal of Engineering, Design and Technology, **16**(3), 478-500, 2018, doi.org/10.1108/JEDT-11-2017-0123.

[23] F. Jacobs, R.B. Chase, Operations and supply chain management, McGraw-Hill Education, 2017.

[24] I. Alsyouf, R. Al-Aomar, H. Al-Hamed, X. Qiu, "A framework for assessing the cost effectiveness of lean tools," European Journal of Industrial Engineering, **5**(2), 170-197, 2011, doi.org/10.1504/EJIE.2011.039871.

[25] E.D. Ouden, Innovation design: Creating value for people, organizations, and society, Springer, 2012.

[26] H. S. Kristensen, A. Remmen, "A framework for sustainable value propositions in product-service systems," Journal of Cleaner Production, **223**,

25-35, 2019, doi.org/10.1016/j.jclepro.2019.03.074.

[27] P.G. Patterson, R.A. Spreng, "Modelling the relationship between perceived value, satisfaction and repurchase intentions in a business-to-business, services context: An empirical examination," Service Industry Management, **8**(5), 414-434, 1997, doi.org/10.1108/09564239710189835.

[28] C. Homburg, A. Hoppe, R. Schick, "Accounting for preference dependency in target costing - a note," Quantitative Finance and Accounting, **57**, 845-858, 2021, doi.org/10.1007/s11156-021-00962-9.

[29] H. Miyakawa, N. Kuratomo, H. E. B. Salih, K. Zempo, "Auditory Uta-KARUTA: Sonificated card game towards inclusive design," in The 32nd Annual ACM Symposium on User Interface Software and Technology, 90-92, 2019, doi.org/10.1145/3332167.3357125.

[30] S. Yanagihara, H. Koga, "Differences in human and AI memory for memorization, recall, and selective forgetting," Societal Challenges in the Smart Society, 371-384, 2020.

[31] T. Saito, A four-letter idiom karuta that can be learned aloud by Takashi Saito (Saito Takashi no koe ni dashite oboeru yozizyukugo karuta) (in Japanese), Gentosha, 2021.

[32] L.M. Jiang, "A short visual history of abstraction in early modern Japanese Karuta: simplification, reinterpretation, and localization," Journal of Asian Humanities at Kyushu University, **7**, 61-83, 2022, doi.org/10.5109/4843130.

[33] L.M. Jiang, "Sacralizing the playful secular: The deity of Karuta-gambling at the Nose Kannon Hall in Sannohe, Aomori," Arts, **13**(27), 1-16, 2024, doi.org/10.3390/arts13010027.

[34] R. Kitagawa, T. Itoh, "Visualization of swiping motion of competitive Karuta using 3D bone display," in 27th International Conference Information Visualization (IV), 346-351, IEEE, 2023, doi: 10.1109/IV60283.2023.00065.

[35] T. Kamei, W. Itoi, F. Kajii, C. Kawakami, M. Hasegawa, T. Sugimoto, "Six month outcomes of an innovative weekly intergenerational day program with older adults and school‐Aged children in a Japanese urban community," Japanese Journal of Nursing Science, **8**(1), 95-107, 2011, doi.org/10.1111/j.1742-7924.2010.00164.

[36] H. Azimah, "Effect of using Karuta cards on students' ability to comprehend vocabulary (Experimental research for the seventh grade at Sabilul Mukminin Boarding School Binjai)," Jurnal Pendidikan Indonesia, **5**(5), 177-184, 2024, doi.org/10.59141/japendi.v5i5.2772.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 9, No. 6, 30-46 (2024)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

On Adversarial Robustness of Quantized Neural Networks Against Direct Attacks

Abhishek Shrestha*, Jürgen Großmann

Fraunhofer-Institut für Offene Kommunikationssysteme FOKUS, System Quality Center (SQC), Critical Systems Engineering, Berlin, 10589, Germany

ARTICLE INFO

Article history:
Received: 14 October, 2024
Revised: 10 December, 2024
Accepted: 11 December, 2024
Online: 23 December, 2024

Keywords:
Deep neural networks
Quantization
Adversarial attacks

ABSTRACT

Deep Neural Networks (DNNs) prove to be susceptible to synthetically generated samples, so-called adversarial examples. Such adversarial examples aim at generating misclassifications by specifically optimizing input data for a matching perturbation. With the increasing use of deep learning on embedded devices and the resulting use of quantization techniques to compress deep neural networks, it is critical to investigate the adversarial vulnerability of quantized neural networks.
In this paper, we perform an in-depth study of the adversarial robustness of quantized networks against direct attacks, where adversarial examples are both generated and applied on the same network. Our experiments show that quantization makes models resilient to the generation of adversarial examples, even for attacks that demonstrate a high success rate, indicating that it offers some degree of robustness against these attacks. Additionally, we open-source Adversarial Neural Network Toolkit (ANNT) to support the replication of our results.

1. Introduction

This paper builds upon our recent work, presented at the 5th ACM/IEEE International Conference On Automation of Software Test [1], which involved a comprehensive study on transferability of adversarial examples among quantized networks under various conditions. In this study, we advance the analysis by examining the efficiency of adversarial attacks on quantized networks when attacks are created and applied on the same network (direct attacks). Together, our previous and current work provide a more complete understanding of how quantization affects network vulnerability by addressing both transfer-based and direct attack scenarios.

Adversarial examples are images with deliberately added perturbations which can cause a network to misclassify the image at a high rate [2]. These perturbation vectors are computed using specific algorithms and often distort an image in such a way that it looks benign or clean to human observers but are enough to cause a network to misclassify the image [3, 4].

As the use of DNNs proliferates over various safety-critical domains like medical diagnosis [5], railway [6], and aviation [7], the possibilities of adversarial examples coercing a network into making adversary-controlled decisions become a severe threat. One of the domains where deep learning is rapidly gaining popularity is the embedded systems. For instance, autonomous systems use AI for decision-making by processing sensory information [8], mobile devices use them for image processing [9], and surveillance systems use them for biometric analysis [10]. However, the implementation of deep learning on edge devices is challenging. While the embedded devices are inherently constrained in terms of memory and power resources [11, 12], the state-of-the-art capabilities of DNNs come at a price of tremendous computational power required for running them. A pre-trained neural network comes with a large number of parameters: AlexNet [13] has 60 million parameters; VGG16 [14], an improvement over AlexNet, has 138 million parameters; similarly, ResNet50 [15], another popular DNN, has 26 million parameters. The presence of these large number of parameters mean that the computational demand at run-time is very high and requires the systems implementing these models to have considerable computing capabilities for a smooth operation.

One of the solutions to the limited resource problem is using a high-performance server that handles the deep learning tasks, with the devices just having to communicate with the server. Another solution, which is more widely adopted, is to deploy optimized versions of a base model on the device itself. On-device deployments has several benefits [16]:

- No need to communicate with the server frequently which saves energy.

*Corresponding Author: Abhishek Shrestha, Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany, abhishek.shrestha@fokus.fraunhofer.de

- Privacy is maintained as the user data does not leave the device.
- Performance is improved as round-trips to the server is avoided.

Various methods have been developed to optimize models for on-device deep learning [17, 18, 19]. One such effective approach is model compression through quantization [20], which reduces both the computational complexity and memory footprint of a network by lowering the precision of its values from the default 32-bit floating point (float32) to smaller bitwidths.

However, recent studies have shown that quantized networks remain vulnerable to adversarial examples [21, 22]. The adversarial vulnerability is especially concerning in compressed networks because of the wide-spread use and accessibility of embedded devices as compared to the full-scale networks running on high-end servers. Moreover, adversarial examples are found to be transferable [4, 23, 24, 25]. Samples created in one network (source) are found to be effective when applied on another network (target) trained to perform similar tasks. Our prior work [1] investigated this transferability property. Interestingly, we observed that iterative attacks like the Boundary Attack [26] and Carlini-Wagner (CW) attack [27] showed high efficiency in direct attack settings (where the source and target network are same), even when the networks were quantized. In this paper, we present an in-depth analysis of this behaviour, offering further insights into attack effectiveness on quantized networks.

Our contributions with this work are as follows:

- We consider diverse adversarial attack algorithms to assess the adversarial vulnerability of quantized networks against direct attacks. Our analysis shows that even though some attacks succeed with high rate, quantized networks, in general, offer some resistance against both gradient-based and gradient-free attacks as they require higher distortion to become effective, making samples easier to detect.
- We introduce the Adversarial Neural Network Toolkit (ANNT), a holistic application that streamlines the entire process—from training full-precision and quantized models to generating adversarial examples and evaluating robustness—within a single tool. ANNT, together with the trained models and adversarial images provided with this paper, enables the replication of our results—both from this study and our previous work [1]. Furthermore, ANNT can serve as a valuable resource for the research community, simplifying experimentation by allowing users to train quantized models and immediately test their robustness using various adversarial attacks without switching between tools.

2. Scope of the Study

Only untargeted misclassifications are considered. This means, for an attack algorithm, classification to any class other than the true class is considered as a successful attack. Targetted misclassifications that require attack algorithms to cause misclassifications to a specific target class selected by an adversary are not considered.

When quantizing a network, both activation and weight values are quantized to the same bitwidth. Quantization of activation and weights individually to different bitwidths is possible and could be a subject of further work. Moreover, gradients and bias values are not quantized.

Further, the study is limited to only image classifiers. Datasets, attack algorithms, and DNNs are selected accordingly.

3. Background

3.1. Deep Neural Network (DNN)

A DNN can be defined as a function that maps a high-dimensional input to a vector[1] output. More specifically, a DNN is a classification function that can be expressed as:

$$f(\mathbf{x}, \theta) = \mathbf{y} \tag{1}$$

Here, $\mathbf{x} \in \mathbb{R}^m$ is an input of m dimensions, θ represents parameters (weights and biases) learned during training, and $\mathbf{y} \in \mathbb{R}^n$ is a vector representing probability distribution over n classes, meaning that $y_1 + y_2 + y_3 + \ldots + y_n = 1$ and $0 \leq (y_i)_{i=1}^n \leq 1$. Each y_i in $\mathbf{y}$ represents the probability that the input $\mathbf{x}$ is assigned to class i.

Thus, the class assigned to the input $\mathbf{x}$ is determined by the index of the maximum value in the output vector $\mathbf{y}$. Hence, $y_i = f_i(\mathbf{x})$ being i^{th} output of the network, the output label y is given by:

$$\underset{i}{\operatorname{argmax}}\, f_i(\mathbf{x}) = y \tag{2}$$

The network learns by iteratively adjusting θ based on an optimization algorithm that guides the adjustments by moving in the direction opposite to the loss gradient $\nabla_\theta J(\mathbf{x}, y, \theta)$, where $J(\mathbf{x}, y, \theta)$ represents loss function used to train the network. The gradient $\nabla_\theta()$ is computed with respect to the current network parameters θ. In our work, since we use trained networks, θ is constant (therefore ignored in Equation 2).

3.2. Distance Metrics

Various distance metrics can be used to measure the similarity (or dissimilarity) between the benign and adversarial samples. L_p-norm distances are widely used as one of the performance metrics when generating adversarial examples [26, 27, 28].

Let, $\mathbf{x}^{adv} \in \mathbb{R}^m$ be the corresponding adversarial example of a benign sample $\mathbf{x}$, L_p distance between $\mathbf{x}$ and $\mathbf{x}^{adv}$ for $p \in [0, \infty)$ is given by:

$$|\mathbf{x} - \mathbf{x}^{adv}|_p = \left(\sum_{i=1}^m |\mathbf{x}_i - \mathbf{x}_i^{adv}|^p\right)^{\frac{1}{p}} \tag{3}$$

The L_p-norm distances include:

- L_0 distance (Hamming distance): L_0 counts the number of non-zero elements in $|\mathbf{x} - \mathbf{x}^{adv}|_0$, that is, $|\{\mathbf{x}_i - \mathbf{x}_i^{adv} \neq 0\}|$.

[1] The bold letterings indicate that the corresponding values are vector quantities.

When considering image classifiers, each element of the input vector $\mathbf{x}$ is a pixel value, and thus, L_0 basically counts the number of pixels that have altered between $\mathbf{x}$ and $\mathbf{x}^{adv}$ [27, 28].

- L_1 distance (Manhattan distance): From Equation 3, L_1 distance can be expressed as:

$$|\mathbf{x}-\mathbf{x}^{adv}|_1 = \sum_{i=1}^{m} |\mathbf{x}_i - \mathbf{x}_i^{adv}| \tag{4}$$

- L_2 distance (Euclidean distance): As per Equation 3, the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}^{adv}$ is given by:

$$|\mathbf{x}-\mathbf{x}^{adv}|_2 = \sqrt{\sum_{i=1}^{m} (\mathbf{x}_i - \mathbf{x}_i^{adv})^2} \tag{5}$$

L_2 can remain small even where there are minute changes in many pixels [27].

- L_∞ distance (Chebyshev distance): This is given by:

$$|\mathbf{x}-\mathbf{x}^{adv}|_\infty = max(|\mathbf{x}_i - \mathbf{x}_i^{adv}|_{\{i=1,\ldots,m\}}) \tag{6}$$

Thus, L_∞ measures the largest change in pixel values. This can be used to set a maximum limit up to which a pixel value is allowed to change. While any number of pixels can be modified, each pixel can only be modified to this limit.

3.3. Adversarial Examples

Adversarial examples are generated by adding computed perturbations to a clean image, resulting in distorted samples that look almost identical to the original image to human observers, but cause significant changes in the output class probabilities of a classifier, leading to a misclassification in majority of cases [2, 4, 25, 28]. Adversarial samples are crafted at test time and do not require an adversary to have any kind of influence on the training process [29, 30].

If y^{true} be the true label corresponding to a clean image $\mathbf{x}$, then from Equation 2 we have: $\text{argmax}_i f_i(\mathbf{x}) = y^{true}$. If a perturbation vector $\eta \in \mathbb{R}^m$ is added to input $\mathbf{x}$, resulting in a perturbed example $\mathbf{x}^{adv}$ causing successful misclassification, then:

$$\underset{i}{\text{argmax}}\, f_i(\mathbf{x}^{adv}) \neq y^{true} \tag{7}$$

It is also worth noting that not all adversarial examples cause misclassification. These samples have high probability of causing misclassification but do not guarantee misclassification [25]. In this view, all samples created from an adversarial examples generation algorithm are adversarial examples but it is possible that not all of them are successful in fooling a network.

A DNN learns by iteratively reducing loss by utilizing optimization algorithms like the gradient descent. In other words, a network is made to converge to a point where the parameters are such that the resulting class probabilities yield low loss. With this in mind, the basic concept behind generating adversarial samples is to increase the loss such that the class probabilities are manipulated in a way desired by the adversary. Since it is not possible to modify the network parameters θ at test time, the input itself is varied till the goal of misclassification is met. Thus, for generating adversarial examples, the optimization problem becomes:

$$\begin{aligned} \max_{\mathbf{x}^{adv}} \quad & J(\mathbf{x}^{adv}, y, \theta) \\ \text{s.t.} \quad & |\mathbf{x}-\mathbf{x}^{adv}|_p \leq \varepsilon \end{aligned} \tag{8}$$

Where, $\mathbf{x}^{adv} = \mathbf{x} + \eta$ and ε is the maximum allowed perturbation measured in terms of $|\cdot|_p$ utilized by the algorithm to generate the sample.

3.4. Crafting Algorithms

Inducing misclassification through random perturbations is notably more challenging [23] and therefore definite algorithms are required to compute perturbation vectors of specific magnitude and direction. Usually, these algorithms aim to solve the optimization problem in Equation 8. In our original work [1], we considered five conceptually different algorithms to create such attacks. In this work, we use the same algorithms as we want to study the attack efficiency of these attacks on the source network.

Fast Gradient Sign Method (FGSM) [4]: The attack is based on the reasoning that all non-linear models are trained to behave rather linearly to make the training process easier. For instance, commonly used activation functions like ReLU are piecewise linear and even sigmoid functions are tuned to work within the linear part of the curve. As a consequence, adding linear perturbations to the input can break the models.

If perturbation vector η be the distortion introduced to input vector $\mathbf{x}$ such that $\mathbf{x}^{adv} = \mathbf{x} + \eta$. For $|\eta|_\infty < \varepsilon$, where ε is less than the precision of the model, the model should not respond to this distortion. However, for a linear model with weight vector $\mathbf{w}$, this distortion grows by $\mathbf{w} \cdot \eta$ as shown by the relation in Equation 9.

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}^{adv} &= \mathbf{w} \cdot \mathbf{x} + \mathbf{w} \cdot \eta \\ \mathbf{w}^T \mathbf{x}^{adv} &= \mathbf{w}^T \mathbf{x} + \mathbf{w}^T \eta \end{aligned} \tag{9}$$

To maximize the effect of this distortion, direction of max-norm constrained perturbation η can be aligned with weight vector. Then, m being the average weight of each element of $\mathbf{w}$ and n be the dimension of $\mathbf{w}$, the activation change can be represented as in Equation 10

$$\mathbf{w}^T \eta = \varepsilon mn \tag{10}$$

The consequences of Equation 10 are: (a) Keeping the average weight same, change in activation due to η grows linearly with n. Thus, a small change at input can aggregate to create large change in output at high dimensions. (b) Since all models behave linearly, the concept of this linear perturbation can also be applied to DNNs to cause misclassifications.

Authors then use these ideas to propose FGSM which adds linear distortion to the input in a single step to create adversarial samples. The perturbation vector η is constructed as:

$$\eta = \varepsilon sign(\nabla_{\mathbf{x}} J(\mathbf{x}, y, \theta)) \tag{11}$$

Thus, the adversarially perturbed sample is given by:

$$\mathbf{x}^{adv} = \mathbf{x} + \varepsilon sign(\nabla_{\mathbf{x}} J(\mathbf{x}, y, \theta)) \tag{12}$$

As can be seen, the perturbation is added in the direction of the loss gradient computed with respect to input $\mathbf{x}$. This makes sense because loss gradient gives the direction of the largest increase in loss. Thus, perturbation aligned with this direction is optimal for increasing loss. ε is the L_∞ norm of the perturbation which also gives the distance between $\mathbf{x}$ and $\mathbf{x}^{adv}$.

Jacobian Saliency Map based Attack (JSMA) [29]: The attack generates adversarial examples by establishing a direct relationship between input variations and output changes, allowing it to identify the features most effective in altering the classifier's decision.

The basic idea behind the algorithm can be summed up in three steps:

1. Compute forward derivative of the function learned by the network to create a mapping between rate of change in output with respect to change in input.
2. Create a saliency map based on the computed forward derivative to search for the most sensitive features that produce change towards the adversarial class.
3. Add defined perturbation to the selected features. Keep adding changes iteratively with each iteration computing the forward derivative and the saliency map until the misclassification is achieved.

The forward derivative of a network computes how much the output $\mathbf{y}$ changes due to the change in $\mathbf{x}$. Since a network learns a vector valued function, the forward derivative has to compute change in each element of $\mathbf{y}$ due to change in each element in $\mathbf{x}$. This is basically the Jacobian of the vector valued function learned by the network.

Thus, the forward derivative is given by:

$$\nabla f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \left[\frac{\partial f_j(\mathbf{x})}{\partial x_i}\right]_{i\in 1,\ldots,m, j\in 1,\ldots,n} \tag{13}$$

For more clarity, when assuming a 2-dimensional input $\mathbf{x}$ and output $\mathbf{y}$, the forward derivative is computed as:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} \end{bmatrix} \tag{14}$$

In the Jacobian matrix, a positive rate of change of an output class means that the change in the corresponding input feature will increase its current prediction probability, while a decrease means that it will decrease its prediction probability. Based on this, a saliency map can be constructed which filters the features that are most important based on the given criteria. Equation 15 provides a very basic filter criteria as defined in [29].

$$S(\nabla f(\mathbf{x}),t)[i] = \begin{cases} 0 \text{ if } \frac{\partial f_t(\mathbf{x})}{\partial x_i} < 0 \text{ or } \sum_{j\neq t} \frac{\partial f_j(\mathbf{x})}{\partial x_i} > 0 \\ \left(\frac{\partial f_t(\mathbf{x})}{\partial x_i}\right)\left|\sum_{j\neq t} \frac{\partial f_j(\mathbf{x})}{\partial x_i}\right| \text{ otherwise} \end{cases} \tag{15}$$

Here, t is the target class to which the input is to be misclassified, that is, $t \neq y^{true}$. $S(\nabla f(\mathbf{x}),t)[i]$ is the saliency map computed for i^{th} feature.

Thus, as per Equation 15, from the Jacobian matrix, features that increase the target class probability and at the same time decrease the probabilities of all other classes are weighed. The feature with the highest value is then selected. In each iteration, selected feature is perturbed by a defined amount. This is continued till misclassification is achieved, that is, $\text{argmax}_i f_i(\mathbf{x}) = t$.

In practice, saliency map criteria as defined in Equation 15 is too restricting; thus, an optimized version which selects a pair of features in one iteration is often used [29]. The policy for generating maps may need to be optimized as per requirement. For instance, pairwise selection is usable for CIFAR10 [31] and MNIST [32] datasets but was found to not work for datasets that contain high-resolution images like the ImageNet [27].

Universal Adversarial Perturbation (UAP) [3]: UAP is different from other attacks discussed in this section as it generates image-agnostic perturbations. Instead of computing adversarial images for each image in a dataset, the algorithm aims to find a single perturbation vector from a given subset of data which can then be applied to the entire data distribution to create adversarial samples. These types of perturbations are called Universal Adversarial Perturbations (UAP). The prefix *universal* is used because they are generalizable across new data points that were not used when creating the perturbation.

If $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a dataset sampled from a data distribution μ, then the goal is to compute a universal perturbation $\boldsymbol{v} \in \mathbb{R}^m$ using X, such that for most $\mathbf{x} \in \mathbb{R}^m$ in μ, Equation 16 is fulfilled.

$$f(\mathbf{x}_k + \boldsymbol{v}) \neq y^{true} \tag{16}$$

The algorithm iterates though each image in X and computes a perturbation vector $\Delta\boldsymbol{v}_k$ that sends the current data point $\mathbf{x}_k + \boldsymbol{v}$ across the decision boundary. Perturbation $\boldsymbol{v}$ is then updated as $(\boldsymbol{v} + \Delta\boldsymbol{v}_k)$.

The perturbation vector $\boldsymbol{v}$ is such that $|\boldsymbol{v}|_p < \xi$, where $|\cdot|_p$ is the desired L_p-norm. To make sure that the magnitude of perturbation $\boldsymbol{v}$ is within ξ, updated $\boldsymbol{v}$ is again projected onto a L_p ball of radius ξ centered at 0. The algorithm stops when a pre-defined fooling rate is obtained on X. If δ be the desired accuracy on X then the required fooling rate is denoted by $(1-\delta)$.

At the end of each iteration, the computed perturbation $\boldsymbol{v}$ is added to all data points in X to create a set of perturbed data points $X_v = \{\mathbf{x}_1 + \boldsymbol{v}, \mathbf{x}_2 + \boldsymbol{v}, \ldots, \mathbf{x}_n + \boldsymbol{v}\}$. The current fooling rate is then given by Equation 17. The algorithm stops when $Err(\mathbf{x}_v) \geq (1-\delta)$.

$$Err(\mathbf{x}_v) = \frac{1}{n}\sum_{k=1}^{n} 1_{[f(\mathbf{x}_k+\boldsymbol{v})\neq f(x_k)]} \tag{17}$$

The individual image perturbation $\Delta\boldsymbol{v}_k$ can be computed using any algorithm. For instance, authors in [3] use DeepFool [33], while [34] uses Projected Gradient Descent (PGD) [35]. In our case, we use FGSM to compute this vector and measure ξ in L_∞.

The Carlini-Wagner (CW) Attack [27]: Carlini and Wagner introduce three variants of one of the most powerful gradient-based adversarial attacks against neural networks. The attacks not only cause misclassification with high success rate but they do so while introducing comparatively low distortion than other attacks like FGSM and JSMA. The three variations of the attacks are based on the L_2, L_0, and L_∞ distance metrics. However, as in our previous

work [1], we only focus on the L_2 version as it is considered to be the strongest [27].

The effectiveness of the attack can be attributed to the optimization problem (Equation 18) that balances two objectives: (1) Minimize distance between adversarial and the original image. (2) Misclassify the image into any class other than the original. This results in adversarial samples that are minimally perturbed while ensuring misclassification.

$$\begin{aligned} &\text{minimize } \|\mathbf{x}^{adv} - \mathbf{x}\|_2^2 + c \cdot l(\mathbf{x}^{adv}); \text{ where,} \\ l(\mathbf{x}^{adv}) &= \max\left(Z_i(\mathbf{x}^{adv}) - \max\left(Z_t(\mathbf{x}^{adv}) : t \neq i\right) + \kappa, 0\right) \end{aligned} \tag{18}$$

In equation 18, $Z_i(\mathbf{x}^{adv})$ is the logit corresponding to the true label and $Z_t(\mathbf{x}^{adv})$ corresponds to any other label $t \neq i$. The equation considers L_2 norm. The amount of distortion or the misclassification confidence can be controlled by varying κ. Larger κ means samples are more stronger (high confidence misclassification) at the cost of higher distortions. On the other hand, c is a positive constant that mediates the trade-off between minimizing perturbation and achieving misclassification. It is determined via binary search during the attack.

Further, the attack considers logits rather than activation values from the softmax layer, this enables the attack to still be effective on networks that apply gradient-masking techniques like the defensive distillation [36]. Moreover, the original paper designs the attack for targeted misclassifications, adapting the objective function accordingly. In this work, we focus on untargeted misclassification and therefore utilize the objective function presented in Equation 18.

The Boundary Attack (BA) [26]: The Attack uses model's decisions on the input points to craft adversarial examples and therefore, unlike other attacks discussed in this section, it does not require access to model parameters or architecture to create adversarial samples.

For each clean image, the algorithm initializes a random adversarial image and iteratively applies perturbations that reduce the L_2 distance between the adversarial and the corresponding clean image. After each iteration, the algorithm checks that the perturbed image remains outside the decision boundary of the original image by querying the model. This process continues until the minimum distance between the clean and adversarial image is achieved.

The algorithm internally uses two parameters, δ and ε to control the perturbations that guide the initialized image towards the clean image. δ controls the magnitude of the perturbations and ε controls the step size towards the clean image. The generation process begins by sampling feature values from a uniform distribution $\mathscr{U}(0,1)$ to create a random image that is adversarial to the clean image. Multiple perturbations are sampled randomly from an iid Gaussian distribution $\mathscr{N}(0,1)$ and are rescaled based on the current value of δ. These perturbations are then projected on a sphere around the clean image and are then added to the random image. From the resulting perturbed images, only those that are still adversarial are selected for further processing. If less than 20% of the perturbed images are adversarial, then this means that the image is already close to the decision boundary, and thus the value of δ is decreased. However, if more than 50% are adversarial, then δ is increased. Finally, to make a movement towards the decision boundary, the perturbations are again scaled by ε and added to the perturbed images. Again, out of the resulting perturbed images, only those that remain adversarial are selected. Value of ε is adjusted by considering similar thresholds as in the case of δ. From the successful adversarial images, the adversarial image that is closest to the initial image in terms of L_2 distance is selected for the next iteration. The loop continues with the updated values of δ and ε until an adversarial image with minimum possible L_2 is obtained. Thus, with each iteration, the adversarial image comes closer to the decision boundary and starts to look like the original image, yet remaining adversarial.

Both δ and ε are adjusted automatically during generation. The number of iterations, however, is provided as input to the algorithm. Fewer iterations result in higher distortion, while more iterations result in less distortion, as the algorithm has more opportunities to bring the initial image closer to the original image.

Moreover, BA is a gradient-free attack as it does not use any type of gradient-information to craft adversarial samples.

3.5. Quantization as a Model Optimization Technique

Quantization reduces the computational complexity during training and inference by reducing the bitwidth of activations, gradients, and weights [18] to lower bitwidth numbers. This allows floating-point multiplications during convolution operations to be replaced with faster bitwise operations. For instance, by binarizing weights and input activations of convolution layers, the dot products during forward pass can be computed with the formula as in Equation 19 [37].

$$\mathbf{x} \cdot \mathbf{y} = \text{bitcount}(\text{AND}(\mathbf{x}, \mathbf{y})), x_i, y_i \in \{0,1\} \forall i \tag{19}$$

Here, $\mathbf{x}$ and $\mathbf{y}$ are two bit vectors and the *bitcount* operation counts the number of 1s in the resulting vector. Equation 19 can be further extended to be valid for any fixed-point integer values. If $\mathbf{x}$ be a sequence of M-bit integers and $\mathbf{y}$ be a sequence of K-bit integers then:

$$\mathbf{x} = \sum_{m=0}^{M-1} c_m(\mathbf{x}) 2^m \tag{20}$$

$$\mathbf{y} = \sum_{k=0}^{K-1} c_k(\mathbf{y}) 2^k \tag{21}$$

where, $(c_m(\mathbf{x}))_{m=0}^{M-1}$ and $(c_k(\mathbf{y}))_{k=0}^{K-1}$ are bit vectors. Then the dot product of $\mathbf{x}$ and $\mathbf{y}$ is given by Equation 22 [37].

$$\mathbf{x} \cdot \mathbf{y} = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} 2^{k+m} \text{ bitcount}[\text{AND}(c_m(\mathbf{x}), c_k(\mathbf{y}))] \tag{22}$$

Thus, by representing activation, weights, and gradients by integer values, convolution operations between them can be greatly optimized.

There are two types of quantization [20, 22]: *post-training quantization* and *quantization aware training*. In post-training quantization, weights and activation values are quantized after a model is fully trained. Quantization aware training quantizes weights, activations, or gradients during training.

DoReFa-Net [37]: The quantization method quantizes weights, activations, and gradients to lower bitwidths during training. Activation and weight values are quantized during forward pass, while the gradients are quantized during backward pass. Although DoReFa-Net is able to perform low bitwidth quantization of gradients, we do not consider gradient quantization in this work. Thus, the convolution operation between weights and activations during forward pass takes place in low bitwidths, while backward pass still requires convolution between quantized and unquantized values.

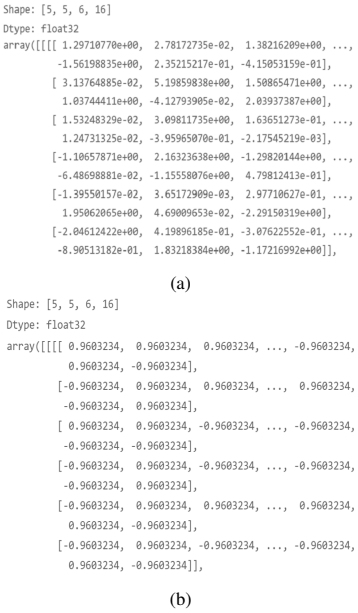


Figure 1: 1-bit quantization of weights using DoReFa-Net: (a) Weight values from a part of a full-precision float32 convolution layer. (b) The same values after 1-bit quantization using DoReFa-Net (without conversion to integers).

If q is a quantized value of p and c is the cost function, then during backward pass, computation as in Equation 23 requires $\frac{\partial q}{\partial p}$ which is not well defined. This creates a problem during back-propagation.

$$\frac{\partial c}{\partial p} = \frac{\partial c}{\partial q} \cdot \frac{\partial q}{\partial p} \tag{23}$$

One of the solution to this problem is to estimate the value of $\frac{\partial q}{\partial p}$, given that $\frac{\partial c}{\partial q}$ is properly defined. These estimators that allow defining custom $\frac{\partial q}{\partial p}$ are called Straight Through Estimators or STEs [38]. DoReFa-Net uses $\mathbf{quantize}_n$ STE [37], which is defined as in Equation 24.

$$\begin{aligned} \textbf{Forward: } & r_o = \frac{1}{2^n - 1}\text{round}((2^n - 1)\ \ r_i) \\ \textbf{Backward: } & \frac{\partial c}{\partial r_i} = \frac{\partial c}{\partial r_o} \end{aligned} \tag{24}$$

Equation 24 uses $\frac{\partial c}{\partial r_o}$ as an approximate of $\frac{\partial c}{\partial r_i}$. In the equation, $r_i \in [0,1]$ is a float32 real number and $r_o \in [0,1]$ is the quantized output value representable by an n-bit number. Since there is always an affine mapping between fixed-point integers and n-bit numbers, the bit-convolutions as specified in Equation 22 can take place between quantized weights and activations during forward pass. This significantly speeds-up the training and inference process.

Figure 1 compares weight values of a convolution layer before and after 1-bit quantization using DoReFa-Net. As illustrated in the figure, the weight values are 1-bit quantized (2 possible values) but the data type remains float32. We maintain the n-bit numbers as float32 and do not convert them to integers. This approach preserves the levelling effect caused due to quantization, but without the speed optimizations that integer representations could provide. However, improving computational speed is not a priority, as the primary goal is to analyze the network's behaviour.

Quantization of weights: DoReFa-Net treats 1-bit quantization of weights differently than n-bit quantization where $n > 1$. For 1-bit quantization, a method similar to [39] is used. The STE is as shown in Equation 25.

$$\begin{aligned} \textbf{Forward: } & r_o = sign(r_i) \times \mathbf{E}(|r_i|) \\ \textbf{Backward: } & \frac{\partial c}{\partial r_i} = \frac{\partial c}{\partial r_o} \end{aligned} \tag{25}$$

Here, $sign(r_i) = 2\mathbb{I}_{r_i>0} - 1$ has two possible values: -1 and 1. $\mathbf{E}(|r_i|)$ is the average of absolute values of all weights in the layer. For n-bit quantization, forward operation as in Equation 26 is used.

$$\textbf{Forward: } r_o = f_w^n(r_i) = 2\text{quantize}_n\left(\frac{\tanh(r_i)}{2\max\left(|\tanh(r_i)|\right)} + \frac{1}{2}\right) - 1 \tag{26}$$

Here, *tanh* bounds the value of r_i within [-1,1]. The expression $\left(\frac{\tanh(r_i)}{2\max\left(|\tanh(r_i)|\right)} + \frac{1}{2}\right)$ results in a value between [0,1], maximum here is taken over all weights in that layer. f_w^n thus quantizes weights to n-bit numbers within [-1,1].

Quantization of activations: The input to each weight layer is quantized with forward operation as defined in Equation 27.

$$\textbf{Forward: } r_o = f_a^n(r_i) = \text{quantize}_n(r_i) \tag{27}$$

Here, r_i is passed through an activation function that limits it within [0,1] before being used as input to f_a^n.

4. Related Work

In our previous work [1], we performed a comprehensive analysis of transferability among quantized and full-precision networks trained on the MNIST and CIFAR-10 datasets. The analysis involved various attack algorithms, as well as variations in model-related properties like architecture and capacity. The findings show that although transferability, in general, remains poor, it may be possible to improve the attack transfer rate using UAP. Further, it was observed that the attacks like BA and CW had high efficiency when applied on the source network even in the case of low-bitwidth networks. Additionally, it was observed that an attacker might be able to predict the success rate of an attack on a target network with different bitwidths, capacities, and architectures based on the performance of the attack when transferred among different bitwidth versions of the source model.

There has been substantial research related to network quantization, adversarial examples, and the impact of adversarial attacks on both full-precision and quantized models, providing significant insights into the vulnerability and robustness of quantized networks.

4.1. Quantization

A survey on various works on quantization of DNNs is presented in [20]. The paper provides an overview of different types and techniques of quantization along with the references to different networks that implement those methods. The case studies presented in the paper involving XNOR-Net [39] and Binaryconnect [40] provide a good starting point for understanding binarized networks.

The paper also provides an introduction to the DoReFa-Net method [37] which is used in this work for quantization. Further, it also compares DoReFa-Net with other quantization methods in terms of accuracy of the resulting quantized networks. The comparisons in this paper helped to confirm that DoReFa-Net had no known issues and that the performance was comparable, if not better, than other similar quantization techniques. This strong performance was a key factor in our decision to select DoReFa-Net for quantization.

Authors in [18] provide details on how TensorFlow Lite [41] can be used for quantization. Although the strategies and the quantization process itself are only focused on TensorFlow Lite's implementation, the findings are significant and can be generalized for other quantization tools as well. The key takeaway is that fine-tuning an already trained network leads to better accuracy models after quantization than training from scratch and that the models with large number of parameters are more resistive to accuracy loss due to quantization. This observation is in agreement with the conclusion drawn from the model configuration experiment in [37].

In [42], authors present an open-source model optimization framework called Mayo which supports multiple compression techniques like the Low-rank Approximation (LRA) [43], quantization and pruning [17]. These compression techniques are implemented through objects called *overriders* which can be applied to any network component like weights, biases, activations or gradients to customize their value. Further, Mayo also allows chaining of multiple overriders meaning that multiple compression techniques can be applied in a sequence. This unique ability enables Mayo to achieve higher compression ratio than any other compression APIs.

However, there are several drawbacks with Mayo; for instance, it uses multiple YAML files for configuration, which makes it customizable but also makes the control flow complex and hard to comprehend for custom implementations. Moreover, there is no clear explanation on how quantization is performed. Authors mention that the quantization is fixed point [42] but do not go into details on how this is done.

The Model Optimization Toolkit[2] from TensorFlow provides multiple methodologies for network quantization. However, the post-training quantization does not support quantization other than 16-bit float and 8-bit integer. The quantization aware training allows to define specific bitwidths for each layer during training, but quantization parameter configuration (like custom bitwidths) are not supported for deployment, meaning that although network layers can be trained at lower bitwidths, model execution takes place at 8 bits. Therefore, lower bitwidth quantization is not possible.

4.2. Adversarial Examples

In [4], authors argue that adversarial examples exist not due to extreme non-linearity or over-fitting of a model, but rather because of its linear behaviour in high dimensions. Authors use this hypothesis to introduce the Fast Gradient Sign Method (FGSM) for creating adversarial examples. FGSM being able to produce successful adversarial examples provides validity to the claim that these examples exploit the model's inherent linearity. The paper also makes an important observation that the adversarial examples exist in broad contiguous regions in input space rather than in fine pockets. For multiple models, these adversarial subspaces are shared. Perturbations leading to the shared subspaces lead to adversarial transfers. Thus, direction of perturbation is important for transferability rather than magnitude.

Authors further explore the concept of adversarial subspaces in [23] where they estimate the dimensionality of this subspace. They find that compared to the input dimension, the dimension of the adversarial subspace is relatively small. The perturbation directions leading to the adversarial subspace are referred to as *adversarial directions*. These orthogonal adversarial directions are shared across multiple models, forming a common subspace. As a result, all adversarial points within this shared subspace are transferable, meaning they can fool any model that share it. Further, authors show that the minimum distance required to cross the decision boundary for any data point is least in the adversarial direction while it is higher in random directions. This means that adding small perturbations is enough to make the data point cross the decision boundary if the perturbation is in the adversarial direction, while larger perturbations are necessary if the perturbation directions are random.

A comprehensive study on how model-specific properties like model accuracy, capacity, and architecture affect transferability is presented in [24]. Here, the authors use Iterative Fast Gradient Sign Method (IFGSM) [25] and FGSM to generate adversarial attacks; hence, the findings are valid only for attacks that leverage loss gradients to create adversarial samples. Authors show that the attacks crafted on low-accuracy networks have very poor transferability regardless of model's capacity and that same architecture transfers are better than different architecture transfers. Further, authors argue that the iterative attacks transfer better than single-step attacks; however, direct attack effectiveness is not considered.

In [30], authors use a custom attack based on Projected Gradient Descent (PGD) algorithm [35] to study both transferability and direct attack effectiveness on various types of networks. Different types of classifiers including Support Vector Machines (SVMs), logistic regression, and neural networks are considered. Authors find that high-complexity networks require less distortion to produce successful adversarial examples because sudden changes in the loss function mean local optima are easier to find. This also meant that highly regularized models were hard to create successful adversarial samples against. Moreover, authors also find that both transferability and attack performance on the source network increases when the hyperparameter value associated with the attack is increased. However, since the attack is gradient-based, the observations, like

[2]https://www.tensorflow.org/model_optimization

[24], are limited for gradient-based attacks.

The study in [44] offers a unique perspective on adversarial examples, arguing that all datasets contain non-robust features which are imperceptible to humans but are highly predictive. Models become sensitive to these features as they learn to rely on them during training. These features are brittle and therefore samples with slight change in them can cause misclassification. Additionally, these features being non-perceptible also means that changes are not visible. The presence of these non-robust features also leads to adversarial examples being transferable because all datasets contain these features and thus models trained on similar datasets are likely to learn similar non-robust features.

In [45], authors present a study on transferability among multiple networks. The paper considers various models including ResNet50, ResNet101, ResNet152, VGG16, and GoogleNet [46]. FGSM, FGM (Fast Gradient Method), and a custom optimization based attack[3] are used to generate adversarial examples. Both FGM and optimization based attack were found to have similar transferability. Interestingly, the transferability between networks with similar architectures was not found to be consistently better than between networks with different architectures, a result that contrasts with [24] but aligns with our findings in [1]. Moreover, the authors observed that the optimization based attack performed better than the other two attacks when applied on the source network, possibly because FGSM and FGM create adversarial samples in a single step and thus sacrifice efficiency for speed.

Authors in [47] use a tool called Deep Learning Verifier (DLV) [48] to generate adversarial examples. DLV performs an exhaustive search within a defined radius around an image and returns all possible adversarial examples (if any) and thus provides a guarantee that apart from the ones that are discovered, the image is robust against all other perturbations within the defined region. In their experiments, authors find that some classes in MNIST dataset had smaller number of effective adversarial samples than others. This indicates that not all classes in a dataset are equally robust and some might be more vulnerable than others.

Regarding adversarial attack generation, there are several popular tools that can be used to implement multiple attack algorithms. For instance, [26] uses FoolBox [49] to implement FGSM and DeepFool [33] attacks; [27] and [23] uses CleverHans library [50] to implement JSMA and FGM, respectively. Moreover, several works like UAP provide open source access to their work[4] so that the community can build on them. In [1] and in this work, we use Adversarial Robustness Toolbox (ART) [51] to create adversarial examples. The library supports comparatively large number of attack algorithms and provides comprehensive documentation for each attack implementation, along with an actively maintained codebase[5].

4.3. Vulnerability of Quantized Networks to Adversarial Examples

In [22], authors use multiple attack algorithms to evaluate the robustness of quantized networks against adversarial attacks. The paper uses DoReFa-Net for network quantization. However, Binary Neural Network (BNN) [52] is used for 1-bit quantization while DoReFa-Net is used only for 2-bit, 3-bit, and 4-bit quantization. The robustness of quantized networks against attacks created on the same network as well as against transfer-based attacks is examined for five attack types: FGSM, Basic Iterative Method (BIM) [25], Simultaneous Perturbation Stochastic Approximation (SPSA) [53], CW attack, and Zeroth Order Optimization (ZOO) [54]. Authors observe that gradient masking caused by activation quantization may increase the robustness of a quantized network against gradient-estimation algorithms like ZOO and gradient-based attacks like FGSM and BIM, but some gradient-estimation algorithms, like SPSA and CW, that are specialized to handle noise function were found to be still effective. These observations are similar to ours as we discuss in Section 7.2. However, we analyse this phenomena further with additional attacks. Furthermore, authors show that weight-only quantization does not affect attack performance at source as the attacks can still produce similar variance in logit values in quantized networks as in full-precision networks.

The work in [21] investigates the transferability of adversarial attacks among compressed networks. The study considers pruning and quantization as compression techniques and uses Mayo [42] for compression. BIM, IFGM, and DeepFool are used to craft adversarial examples. Authors show that the density of a network can be reduced to as low as 15% for both CIFAR10 and MNIST networks without any reduction in the test accuracy. This is interesting because it shows that majority of parameters in a network are not significant and supports the claim in [20] that low-bitwidth quantization works because most parameters in a network are not useful.

Authors in [34] present a transferability study that implements various compression techniques including quantization to analyse the transferability of the UAP attack across compressed networks. A method called Additive Powers-of-two (APoT) [55] is used for quantization. PGD is used to create UAPs. The difference between the UAP crafted using PGD and FGSM (as used in this work) is that the PGD updates the overall noise vector $\boldsymbol{v}$ in mini-batches, while FGSM updates it per image. An important observation is that SVHN dataset [56] was found to be more robust against attacks created and applied on the same network as compared to CIFAR10 even when both CIFAR10 and SVHN were trained on the same network and images in both datasets had the same resolution. This indicates that some datasets are more robust to adversarial attacks due to the nature of data. Further, similar to [22], authors argue that quantization can cause gradient-based attacks to perform poorly on the source network due to gradient masking. Experiments in the paper also show that transferability is poor when source and target networks differ in bitwidths, as well as when they differ in the type of compression algorithm used.

5. Adversarial Neural Network Toolbox (ANNT)

We introduce Adversarial Neural Network Toolbox (ANNT) [57], an open-source tool that provides a unified interface for handling

[3] Optimization based attack implemented by the authors iteratively adds perturbation to a clean sample until the loss is large enough to cause misclassification.
[4] https://github.com/LTS4/universal
[5] https://github.com/Trusted-AI/adversarial-robustness-toolbox

the complete workflow of quantized model training, adversarial image creation, and model robustness evaluation. Figure 2 shows the usability of the tool in terms a basic workflow.

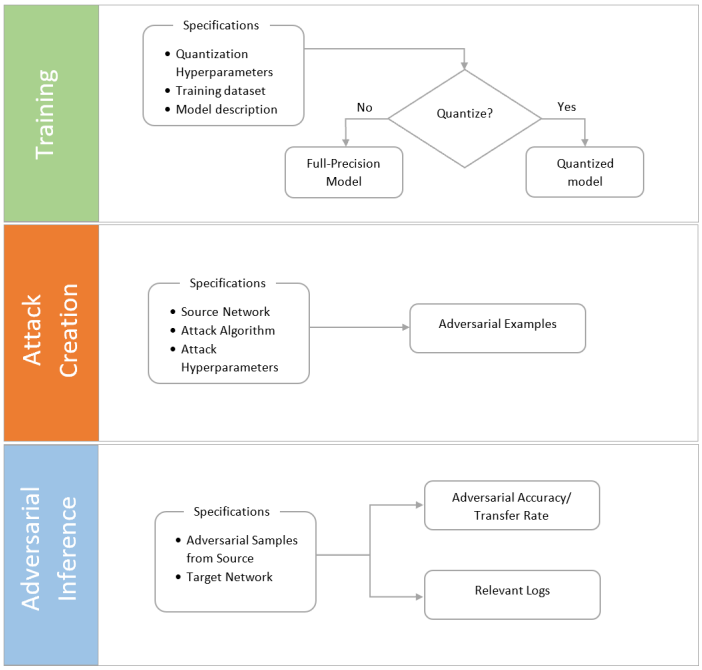


Figure 2: The custom API can be used to train models, create adversarial examples and transfer created adversarial examples.

To train full-precision models, users can provide the model description (architecture), input dataset, and training hyperparameters. Additionally, quantization hyperparameters such as the weight, activation, and gradient bitwidths can be provided to train quantized versions of a network.

The tool provides a unique functionality of generating adversarial examples on a network of specified bitwidth. Users can provide a trained model, quantization bitwidth, adversarial attack algorithm, and attack hyperparameters to create specified number of adversarial samples.

Further, the tool can also be used to perform adversarial inferences on any given target network. The process computes the accuracy of the target network against the input adversarial samples. It also generates other relevant information like the correctly and incorrectly classified samples and average L_∞ and L_2 distance between the successful adversarial and clean samples. Thus, the cumulative information generated is enough to perform a comprehensive analysis of both direct and transfer-based attacks.

The tool can be used as a standalone Python application or as a library. When used as an application, configurations like current task (training/ inference/ attack creation), bitwidths, adversarial attack algorithm (for attack creation), training hyperparameters, and dataset can be provided through a YAML file. The tool then executes the specified task and provides detailed logs of the entire process. When used as a Python library, users can simply import ANNT as a module and utilize the provided interfaces for each task. Additionally, an interface to load the generated samples for visualization is also included. The repository provides a detailed wiki as well as sample notebooks to help users get started with the tool.

In addition to MNIST trained LeNet-5 [32] and CIFAR10 trained Resnets [15] of different capacities including Resnet20, Resnet32, and Resnet44, several custom Convolutional Neural Networks (CNNs) trained on MNIST and CIFAR10 are supported out-of-the-box. Further, five different attack types—FGSM, CW attack, Boundary Attack, JSMA, and UAP—are supported.

The tool is based on TensorFlow 1.13 [58] and uses Tensorpack 0.11 [59] for model training and inference. Further, DoReFa-Net [37] is used for quantization while Adversarial Robustness Toolbox (ART) [51] is used to create adversarial samples.

6. Experimental Setup

6.1. Datasets and Models

The details regarding the datasets and full-precision (32-bit) models used in the experiments are shown in Tables 1 and 2, respectively.

Table 1: MNIST and CIFAR10 datasets.

Dataset	Remarks
MNIST	• 60,000 images in training set, • 10,000 images in test set, • 28x28 grayscale images, • 10 distinct labels
CIFAR10	• 50,000 images in training set, • 10,000 images in test set, • 32x32 colour images, • 10 distinct labels

Table 2: Full-precision (FP) MNIST and CIFAR10 models used in the experiments.

Dataset	Model ID	Test Set Accuracy	Parameters
MNIST	Mnist A	0.991	414K
CIFAR10	Resnet20	0.892	269K

All models were trained from scratch. The MNIST model (named as Mnist A) is a custom CNN while Resnet20 is a ResNet [15] trained on CIFAR10. The model architecture for Mnist A[6] and Resnet20[7] are based on the examples defined in the Tensorpack repository [59].

6.2. Quantization

1-bit, 2-bit, 4-bit, 8-bit, 12-bit, and 16-bit quantized versions of the models in Table 2 were trained. As recommended in [37], the first and last layers were not quantized in favour of better accuracy.

[6]https://github.com/tensorpack/tensorpack/blob/master/examples/basics/mnist-convnet.py
[7]https://github.com/tensorpack/tensorpack/blob/master/examples/ResNet/cifar10-resnet.py

Quantization here refers to weight and activation quantization. Thus, an 8-bit network means both weights and activations are quantized to 8 bits.

Table 3 shows the accuracy of the quantized versions of the Mnist A and Resnet20 models. Like the FP versions, all quantized models were trained from scratch. As can be seen, quantization did not result in noticeable drop in accuracy for models trained on MNIST, while CIFAR10 models show a non-negligible decrease in accuracy. This was expected because DoReFa-Net is known to result in accuracy drops for more natural datasets [37].

Table 3: Test set accuracy of the quantized versions of the Mnist A and Resnet20.

Quantization Bitwidth	Test Set Accuracy *Mnist A*	*Resnet20*
1	0.991	0.834
2	0.991	0.865
4	0.992	0.847
8	0.992	0.829
12	0.991	0.843
16	0.990	0.842

6.3. Attacks and Metrics

Attacks: Table 4 summarizes the attack algorithms used along with other relevant information.

Table 4: Adversarial attack algorithms used and their key characteristics.

Algorithm	Gradient-Based/ Gradient-Free	Iterative/ Single-Step	Distance Metric
FGSM	Gradient-based	Single-step	L_∞
JSMA	Gradient-based	Iterative	L_0
UAP	Gradient-based	Iterative	L_∞
CW	Gradient-based	Iterative	L_2
BA	Gradient-free	Iterative	L_2

Attack hyperparameters: Table 5 shows the selected hyperparameter values for each attack type for both Mnist A and Resnet20 models.

In the case of FGSM, ε controls the magnitude of perturbation introduced to the images. In JSMA, θ is the amount of distortion added per feature in each iteration and γ is the percentage of features allowed to be distorted for an image. For UAP, ε is the perturbation magnitude for FGSM which is used to generate adversarial examples within the UAP (Section 3.4), ξ is the maximum allowed magnitude of perturbation of the UAP noise vector. As recommended in [34], we measure ξ in L_∞. For the Boundary Attack, i is the maximum number of iterations[8]. Finally, for CW attack, $\kappa \geq 0$ controls attack confidence, i is the number of iteration the algorithm runs per image (gradient descent steps), $c > 0$ is a balancing constant used in the optimization problem (Equation 18), b_s is the number of binary search steps to determine c, and c_i is the initial value of c. The values of c_i and b_s were selected based on the original paper [27], while κ was varied to control distortion. The values of hyperparameters in Table 5 were selected such that the images were distorted but yet remained recognizable to human observers.

Table 5: Attack hyperparameter values for the full-precision (FP) and quantized versions of the MNIST and CIFAR10 models.

ModelID	Attack	Hyperparameter	Value
Mnist A	FGSM	ε	0.25
	JSMA	θ	1
		γ (%)	10
	UAP	ε	0.1
		ξ	0.6
	BA	i	15
	CW	κ	5
		i	25
		b_s	20
		c_i	0.01
Resnet20	FGSM	ε	0.05
	JSMA	θ	0.3
		γ (%)	5
	UAP	ε	0.01
		ξ	0.1
	BA	i	12
	CW	κ	5
		i	25
		b_s	20
		c_i	0.01

Attack metrics: Based on the metrics used by the current state of the art, there are two possibilities for representing the effectiveness of an adversarial attack on a network: adversarial accuracy and evasion rate.

Adversarial Accuracy is the accuracy of a network against adversarial examples. It is expressed as the ratio of the number of adversarial examples that are classified correctly by the network to the total number of samples used to attack the network.

For a set of pairs of clean sample and its adversarial counterpart,

$$N = \{(\boldsymbol{x}_1, \boldsymbol{x}_1^{adv}), (\boldsymbol{x}_2, \boldsymbol{x}_2^{adv}), \ldots, (\boldsymbol{x}_n, \boldsymbol{x}_n^{adv})\}$$

the adversarial accuracy is computed as below:

$$\textit{Adv. accuracy} = \frac{\left|\{x^{\text{adv}} \in N : \arg\max_i f_i(x^{\text{adv}}) = y^{\text{true}}\}\right|}{|N|} \quad (28)$$

Here, f is the classifier in which the attack is applied.

Similarly, *evasion rate* gives the success rate of the adversarial attack on a network. It is defined by the ratio of the number of adversarial examples that are classified incorrectly by the network to the total number of samples used to attack the network. This is computed as:

[8]The values of δ and ε, as mentioned in Section 3.4 are adjusted automatically. ART initializes both of them as 0.01, altering this initial value did not create any noticeable change in final quality of samples, and thus were left at their default values for the experiments.

$$\textit{Evasion rate} = \frac{\left|\{x^{\text{adv}} \in N : \arg\max_i f_i(x^{\text{adv}}) \neq y^{\text{true}}\}\right|}{|N|} \quad (29)$$

A network having higher adversarial accuracy means the model is more robust against the attack, while an attack having higher evasion rate means the network is less robust.

Any one of these metrics can be used to represent adversarial robustness. [21, 22, 45] use Equation 28, whereas [23, 24, 30, 34] use Equation 29.

In this paper, we use adversarial accuracy, as we have selected the same metric in [1]. This is simply a preference, using evasion rate would not affect the observations or the results.

Training of all models, adversarial examples creation, and computation of adversarial accuracy on target network were done using ANNT.

7. Experiments, Observations, and Analysis

7.1. Experiments

A random sample of 1,000 clean images was selected from the MNIST dataset (Table 1). Taking FP Mnist A (Table 2), and its quantized counterparts (Table 3) as source networks, adversarial examples were created using all attack types described in Table 4 with attack hyperparameter values as described in Table 5. When creating adversarial examples, inherent inefficiencies of the models were avoided by selecting only those clean samples that were correctly classified by the source network. The samples were then applied on the same source network. The resulting adversarial accuracy of the network, along with the average L_2 and L_∞ distances (as a measure of distortion) between the successful adversarial and clean samples were recorded. The samples were taken again, and the process was repeated for 3 independent runs.

The same procedure was performed for FP Resnet20 (Table 2) and its quantized versions (Table 3). Table 6 shows averaged adversarial accuracies and the L_p distances from the 3 runs for each MNIST and CIFAR10 model.

7.2. Observations and Evaluation

Based on the results in Table 6, the following observations can be made:

Observation 1: The Boundary Attack has high effectiveness. For both CIFAR10 and MNIST models, the Boundary Attack shows very high effectiveness while requiring very less number of iterations to look like the original image. From Figure 3, it can be seen that it just takes about 15 iterations for MNIST and 12 for CIFAR10 for the adversarial images to look like the original image. This also goes along with the observation made in [26] where it takes very less number of iterations to make the initial random image look like the original image with more visible distortions at lower iterations.

The attack's high effectiveness across all models, including the quantized ones, makes sense because it keeps the initialized image adversarial for any number of iterations in all cases.

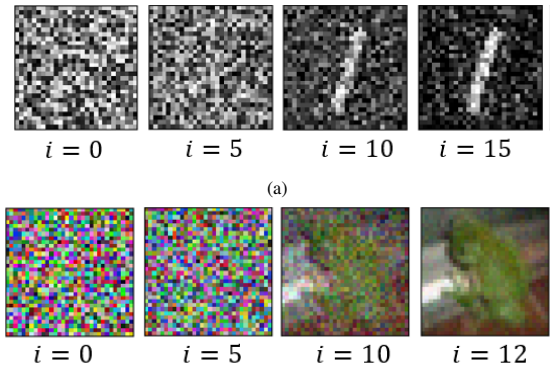


Figure 3: The adversarial image generation progression using the Boundary Attack depicted over multiple iterations on: (a) Mnist A FP model. (b) Resnet20 FP model.

Observation 1.1: Adversarial images generated by the Boundary Attack are more distorted in case of quantized networks. Ideally, when given enough iterations, the Boundary Attack should generate adversarial image which looks exactly like the original image with no visible distortions. However, compared to the FP models, majority of the adversarial images produced with quantized models especially at lower bitwidths were more distorted. This can also be observed in terms of L_2 distances in Table 6 where L_2 distances in case of 1-bit quantized network is higher when compared to the corresponding FP network.

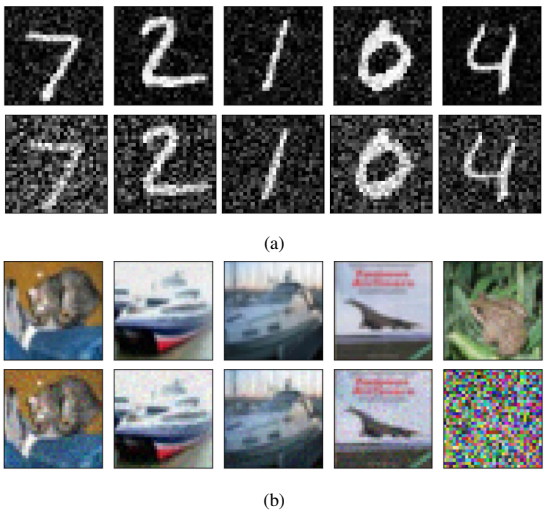


Figure 4: Adversarial examples generated by the Boundary Attack on: (a) Mnist A FP model (top row of 5 images) and 1-bit quantized Mnist A (bottom row of 5 images) for 100 iterations. (b) Resnet20 FP model (top row of 5 images) and 1-bit quantized Resnet20 (bottom-row of 5 images) for 50 iterations. For easier comparison, all image sets are first 5 images from the corresponding datasets.

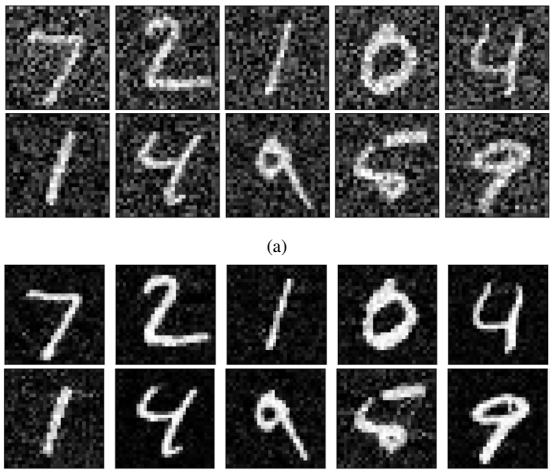
Figure 4 shows a comparison between adversarial images generated from the FP and 1-bit quantized models. As can be seen, the adversarial images for 1-bit models are more distorted with one of the images in the quantized version of Resnet20 being non-recognizable (elaboration in observation 1.2). It can be hypothesized that this is because quantized networks are more resistive to noises in the input data than their FP counterparts. The activation quantization causes activation values to be clipped [21] because of which it

Table 6: The adversarial accuracy of FP and quantized versions of Mnist A and Resnet20 against the five attacks. Attacks were created and applied on the same network. The average L_2 and L_∞ distances between the successful adversarial example and the corresponding clean sample is shown as well.

		Mnist A				ResNet20			
Bitwidth	**Attacks**	**Hyperparameter Values**	**Adversarial Accuracy**	L_2	L_∞	**Hyperparameter Values**	**Adversarial Accuracy**	L_2	L_∞
FP			0.337	5.219	0.25		0.119	2.740	0.05
1			0.845	5.134	0.25		0.137	2.737	0.05
2			0.750	5.116	0.25		0.206	2.742	0.05
4	FGSM	$\varepsilon = 0.25$	0.715	5.128	0.25	$\varepsilon = 0.05$	0.292	2.742	0.05
8			0.678	5.132	0.25		0.299	2.736	0.05
12			0.493	5.120	0.25		0.308	2.736	0.05
16			0.480	5.129	0.25		0.367	2.737	0.05
FP			0.116	5.436	1		0.074	2.425	0.436
1			0.339	7.801	1		0.142	2.504	0.513
2			0.375	6.707	1		0.247	2.581	0.395
4	JSMA	$\theta = 1$, $\gamma = 10\%$	0.140	6.814	1	$\theta = 0.3$, $\gamma = 5\%$	0.419	2.804	0.484
8			0.064	5.739	1		0.351	2.680	0.505
12			0.066	6.370	1		0.469	2.720	0.502
16			0.148	6.981	1		0.430	2.808	0.515
FP			0.114	9.352	0.6		0.176	3.362	0.1
1			0.683	9.073	0.6		0.110	3.430	0.1
2			0.555	8.585	0.6		0.154	3.275	0.1
4	UAP	$\varepsilon = 0.1$, $\xi = 0.6$	0.438	8.685	0.6	$\varepsilon = 0.01$, $\xi = 0.1$	0.169	3.414	0.1
8			0.378	8.648	0.6		0.192	3.308	0.1
12			0.174	8.618	0.6		0.297	3.390	0.1
16			0.162	8.159	0.6		0.175	3.275	0.1
FP			0.037	3.655	0.888		0.000	0.111	0.014
1			0.526	5.220	0.944		0.000	0.822	0.104
2		$\kappa = 5$,	0.558	5.047	0.928	$\kappa = 5$,	0.000	0.360	0.049
4	CW	$i = 25$, $b_s = 20$,	0.148	3.182	0.803	$i = 25$, $b_s = 20$,	0.000	0.249	0.041
8		$c_i = 0.01$	0.190	3.833	0.897	$c_i = 0.01$	0.001	0.155	0.020
12			0.163	3.650	0.874		0.000	0.102	0.013
16			0.106	3.066	0.780		0.000	0.112	0.013
FP			0.000	5.507	0.629		0.000	2.387	0.155
1			0.000	6.259	0.634		0.012	2.816	0.184
2			0.000	4.649	0.507		0.066	2.603	0.170
4	BA	$i = 15$	0.000	4.338	0.488	$i = 12$	0.045	2.859	0.186
8			0.000	4.267	0.493		0.080	2.904	0.189
12			0.001	3.664	0.432		0.002	3.128	0.203
16			0.000	3.235	0.387		0.080	2.803	0.184

becomes hard to produce differential activations from small changes at the input[9], and thus, even when the input has slight perturbations, quantized networks can correctly classify the image. This is also evident from the data from other attack types where for the same value of attack hyperparameters, FGSM, JSMA and UAP perform comparatively bad when the network is quantized. In the case of the Boundary Attack, this could mean that the algorithm cannot further reduce the distortions in an image because then the quantized network will classify the adversarial example correctly.

This hypothesis was put to test by increasing the number of iterations in the Boundary Attack to 1,000 for the 1-bit quantized Mnist A model. As can be seen in Figure 5a, the adversarial images are still equally distorted for 1-bit quantized version; whereas, the distortions are significantly less even for less number of iterations for the FP models, as seen in Figure 5b. Therefore, it would not matter if the iterations are increased any further because the algorithm will not be able to reduce the distortions due to the network being insensitive to small noises at input.



(a)

(b)

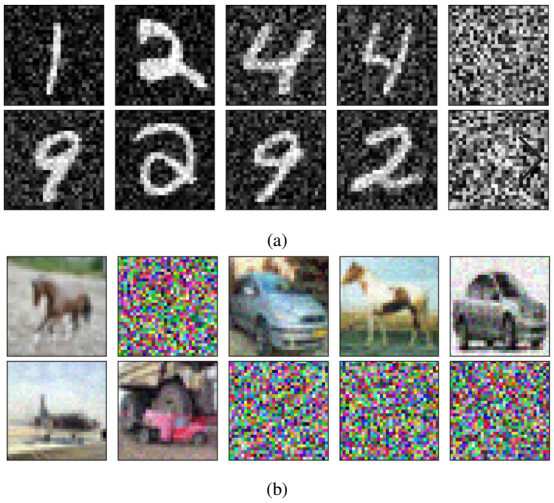
Figure 5: Adversarial examples generated by the Boundary Attack on: (a) 1-bit quantized Mnist A at 1,000 iterations. (b) Mnist A FP model at 200 iterations. Both images are first 10 images from the MNIST dataset.

Quantized networks being more resistive to input noises is also observed in [47] and [34] where the authors find that the perturbations that worked in FP stopped working in quantized networks. Quantization thus acting as a filter for adversarial noise.

Observation 1.2: Imperceptible adversarial images resulting from the Boundary Attack on quantized networks. Apart from the images that are distorted but recognizable to human oracles, the Boundary Attack also resulted in adversarial images that were completely distorted and unrecognizable but only in case of quantized networks. Figure 6 shows adversarial images generated by the Boundary Attack on 1-bit quantized versions of Mnist A and Resnet20 models. As can be seen, multiple images in both figures are unrecognizable.

This could again be due to the algorithm not being able to reduce the distortion any further because of the model being robust against input noises. To verify this, an experiment was performed in which adversarial examples were generated from 3,000 randomly sampled clean images from MNIST and CIFAR10 datasets using the 1-bit quantized versions of Mnist A and Resnet20 as source. The images that seemed to be composed of random pixels were then isolated and inferences were ran on them. It was found that for all of these images, the true class was within top-2 predicted classes. This indicates that the Boundary Attack could not reduce the L_2 distance between the original and the adversarial image any further because any further reduction would cause the image to go inside the decision boundary of the original image making the image no longer adversarial.

Imperceptible images having true labels within top-2 predicted classes also means that although the features in the images in Figure 6 are not recognizable to human observers, the network identifies these features and tries to classify them to the correct class. These features that are non-recognizable to humans but tend to be meaningful and predictive for networks are studied in [44].



(a)

(b)

Figure 6: Adversarial examples generated by the Boundary Attack on: (a) 1-bit quantized Mnist A model at 100 iterations. (b) 1-bit quantized Resnet20 at 50 iterations. The adversarial images were generated from 10 randomly selected clean images from the corresponding datasets.

One of the important qualities of adversarial examples is that they should be classified correctly by human oracles, and since these images are completely distorted, they cannot be considered as adversarial images. Thus, these examples were removed when computing source network performance in Table 6.

It is also worth noting that these imperceptible images are rare. In a set of 3,000 random images, on 3 separate runs, for 1-bit quantized Resnet20 at 12 iterations, only about 280 images on average were imperceptible. Similarly, for 1-bit quantized Mnist A at 15 iterations, on average only about 180 such images were found. Thus, these images formed very small portion of the total adversarial examples generated and only occurred for quantized networks.

The presence of these images when creating adversarial images from the Boundary Attack also suggests that although networks show near-zero resistance against the attack, quantized networks do offer certain form of resilience because valid adversarial images that have less distortion or are at least recognizable to humans become

[9]Weight quantization, on the other hand, does not contribute in poor performance of the attacks when the attacks are crafted in the same network; empirical evidence is presented in [22].

hard to create when networks are quantized.

Observation 2: CIFAR10 models require less distortion than MNIST models to produce misclassification. As can be seen from Table 6, for CIFAR10 models lower distortions are enough for the attacks to perform well, while the MNIST models require comparatively higher value of the attack hyperparameters.

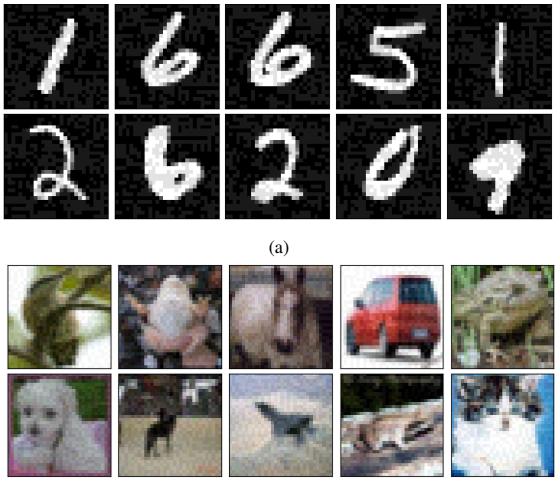


Figure 7: Adversarial examples generated using FGSM on: (a) the FP Mnist A model when $\varepsilon = 0.1$. (b) the FP Resnet20 model when $\varepsilon = 0.025$. The adversarial images were generated from 10 randomly selected clean images from the corresponding datasets.

There are two reasons for this. The first reason is that the MNIST dataset has less intra-class differences [26] which makes the classification problem easier to solve, and thus the network is less sensitive to small changes or perturbations [34]. However, in case of CIFAR10, a single class can have a large variety of objects of different shapes, sizes, and color, and thus a small change in any of the input features is enough to produce misclassification [34]. For instance, considering FGSM with $\varepsilon = 0.1$ in case of FP Mnist A model, the average adversarial accuracy from 3 separate runs on 1,000 samples was found to be 0.831. In contrast, for FP Resnet20 model, for the same attack with $\varepsilon = 0.025$, the average adversarial accuracy was found to be 0.127. Thus, comparatively, CIFAR10 trained network was more vulnerable to the resulting adversarial samples even in relatively low distortion. In both cases, the value of hyperparameter ε is selected such that the distortions were barely visible, as seen in Figure 7. Similar behaviour is reported in [34] with SVHN, an MNIST-like dataset, where SVHN models are more robust to UAP based attacks as compared to CIFAR10 models.

Another reason why CIFAR10 models show more vulnerability is the high-dimensionality of the CIFAR10 dataset as compared to MNIST. For Attacks like FGSM and UAP, which add a constant perturbation to the input in a specific direction, the same value of the constant introduces larger change at the output in higher dimension. This is also evident from Equation 10. Keeping ε constant and increasing n would cause larger change in the activations.

For CW attack as well, we can see that the same value of hyperparameters are more effective in CIFAR10 models as compared to MNIST models.

Observation 3: Quantized models are more robust to loss gradient-based attacks. Quantized models have better adversarial accuracy than their FP counterparts against loss gradient based attacks like FGSM and UAP. Similar behaviour is observed in [21, 22, 34]. The increased robustness can be attributed to the gradient masking caused due to activation quantization. Gradient masking makes the loss surface of the network hard to optimize over[10] [21, 22]. The resulting gradients no longer point to the adversarial examples [22] which makes it harder for these attacks to find useful gradients that can cause misclassification.

In the case of CW attack, high effectiveness can be observed in CIFAR10 models even when the networks are quantized. This is due to the optimization problem (Equation 18) solved by the attack, which, given enough iterations and binary search steps, will lead to misclassification. However, during attack creation, it was harder to craft adversarial samples, especially for lower bitwidths, as the resulting samples were more distorted and took more time to converge. The attack tries to introduce minimal distortion while trying to achieve misclassification (Equation 18), but due to activation quantization, it becomes difficult to achieve this as the network becomes insensitive to small perturbations, especially at lower bitwidths. Hence, even when using logits, where gradients are comparatively more expressive, the attack finds difficulty in converging. This is also evident by the significantly higher L_2 distance in the case of lower bitwidth networks (Table 6) as the attack requires higher values of c and more binary search steps to create samples. Moreover, the table shows that Mnist A has increased L_2 and robustness when quantized, further indicating increased robustness of quantized networks against such attacks.

Observation 4: JSMA performs poorly in quantized networks. The poor performance of JSMA can be explained by how JSMA creates adversarial examples. In each iteration, the JSMA algorithm seeks to find the input features that cause positive change towards the target adversarial class (Equation 15) and at the same time reduce the overall class probabilities of all other classes. When it finds these features, it adds defined amount of distortion to those features (for instance, $\theta = 1$ and $\theta = 0.3$ in Table 6) while also restraining total distortion to a limit ($\gamma = 10\%$ and $\gamma = 5\%$, respectively). When networks are quantized, activation quantization makes the network insensitive to small changes in input as the small noises fail to produce any change in activations. Thus, JSMA struggles to find features that, when distorted by the defined amount, can cause misclassification.

This hypothesis was tested by randomly sampling 2,000 clean samples from MNIST and CIFAR10 datasets and creating adversarial examples using JSMA on Mnist A, Resnet20, and all their quantized versions. Average L_0 distance between the adversarial samples and their corresponding benign counterparts were recorded. Three individual runs were carried out and the average L_0 distance from those runs for each model are as shown in Table 7. As can be seen, on average, quantized networks required more features to be distorted than the corresponding FP model. This indicates that JSMA was struggling to find features to build adversarial examples.

Thus, although not using loss-gradients, the activation quantization causes JSMA to be less effective on quantized networks.

[10]Clipping of activations causes activations to remain in the same bucket causing no change or to switch to another bucket causing large change.

Table 7: Average L_0 distances between the clean and adversarial samples produced using JSMA on the FP and quantized versions of Mnist A and Restnet20 models.

	L_0 **Distance**	
Quantization Level	*Mnist A* ($\theta = 1, \gamma = 10\%$)	*Resnet20* ($\theta = 0.3, \gamma = 5\%$)
FP	50.007	82.592
1	69.047	85.256
2	67.219	102.211
4	55.285	120.840
8	43.437	113.121
12	48.293	121.370
16	53.745	119.264

7.3. Summary

Based on the observations, the following statements can be made:

1. *MNIST models are more robust to adversarial attacks than the CIFAR10 models for some attack types.* FGSM, UAP, CW attack and JSMA were found to be more effective on CIFAR10 models. This can be attributed to the characteristics of the data. MNIST has less variations in a single class, while CIFAR10 has larger variation of objects; thus the classification problem is simpler in case of MNIST as compared to CIFAR10. This is also reflected by the MNIST models being able to achieve very high test accuracies while the test accuracies of the CIFAR10 models are comparatively low (Tables 2 and 3). Further, high dimensionality of CIFAR10 also causes it to have less adversarial robustness.
2. *Quantized networks show resistance against both gradient-based and gradient-free attacks.* Activation clipping causes quantized networks to filter small noises at the input which makes the network more resilient to attacks. This was already known for attacks like FGSM and UAP from the findings in [22] and [34], respectively. This study further demonstrates that this also applies for attacks like JSMA that do not use loss gradients, for search-based attacks like the Boundary Attack, and also for the CW attack that uses logits and a more powerful objective function. Although the Boundary Attack and CW attack depicted very high effectiveness, even on quantized networks, the adversarial samples were found to be more distorted, with the Boundary Attack sometimes producing non-recognizable samples. This can be considered as a form of resilience against the attacks as the samples become more detectable and harder to create. Thus, although limited, quantization seems to provide some resistance against direct adversarial attacks.

8. Discussion

- Attacks like FGSM and UAP, which rely on loss gradients at the output layer to generate adversarial examples, tend to be less effective against quantized networks due to gradient masking [22, 34]. Interestingly, as noted in [22], CW attack demonstrated higher effectiveness in quantized networks, particularly with natural datasets. However, our analysis indicates that quantized networks offer resistance during attack creation. This resistance was also observed with attacks like JSMA and the Boundary Attack, where activation quantization can make networks more robust against direct attacks to some extent.

  Furthermore, the effectiveness of some attack algorithms also depends on the characteristics of the data itself. Models trained on natural datasets like CIFAR10 seem to be more vulnerable to some attacks than those trained on datasets like MNIST. Similar observation was made in [34] for UAP attacks on SVHN and CIFAR10 datasets.

  We consider five different attack algorithms. FGSM is a single-step attack that uses loss gradients to create adversarial examples, whereas JSMA iteratively distorts selected pixels without relying on loss gradient information. UAP, on the other hand, focuses on finding a universal perturbation that can generalize across multiple images, rather than crafting unique adversarial samples for each one. CW attack, in contrast, performs gradient descent towards misclassification. The Boundary Attack is a gradient-free method that generates adversarial samples without requiring access to the model's parameters or training data. Thus, the algorithms are conceptually diverse, allowing the analysis to incorporate a broader range of attack strategies and provide a more comprehensive view on adversarial robustness.

- Reproducibility is a significant challenge in ML. Use of a single tool with well-documented functionality makes it easier for other researchers to reproduce and validate experiments. To facilitate this, we open-source our experimentation tool, ANNT. The consistent interface provided by ANNT for various tasks means that it is easier to standardize experiments and switch between different configurations. Researchers can easily share logs and configurations to replicate experiments.

  The resources used in the experiments in this paper, including trained models, adversarial images, and the experiment results in the form of logfiles (including those from [1]) are available at https://mega.nz/fm/public-links/ql8CwJxb. Thus, the data, along with ANNT is sufficient to replicate the experimental results.

9. Conclusion

In this work, we analyze the adversarial robustness of DNNs under direct attacks. Within this premise, we evaluate multiple attack methods on models trained on CIFAR10 and MNIST datasets and quantized to different bitwidths. Our findings, along with those from [1] indicate that quantization provides some protection against both direct and transfer-based attacks.

We also present ANNT, a tool designed to facilitate the validation of our results and support further research in this area.

References

[1] A. Shrestha, J. Großmann, "Properties that allow or prohibit transferability of adversarial attacks among quantized networks," in Proceedings of the 5th

ACM/IEEE International Conference on Automation of Software Test (AST 2024), AST '24, 99–109, Association for Computing Machinery, New York, NY, USA, 2024, doi:10.1145/3644032.3644453.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, "Intriguing properties of neural networks," CoRR, **abs/1312.6199**, 2014, doi:10.48550/arXiv.1312.6199.

[3] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, "Universal adversarial perturbations," in Proceedings of the IEEE conference on computer vision and pattern recognition, 1765–1773, 2017, doi:10.48550/arXiv.1610.08401.

[4] I. J. Goodfellow, J. Shlens, C. Szegedy, "Explaining and Harnessing Adversarial Examples," arXiv preprint arXiv:1412.6572, 2015, doi:10.48550/arXiv.1412.6572.

[5] M. Yasmin, M. Sharif, S. Mohsin, "Neural Networks in Medical Imaging Applications: A Survey," World Applied Sciences Journal, **22**, 12, 2013.

[6] J. Grossmann, N. Grube, S. Kharma, D. Knoblauch, R. Krajewski, M. Kucheiko, H.-W. Wiesbrock, "Test and Training Data Generation for Object Recognition in the Railway Domain," in P. Masci, C. Bernardeschi, P. Graziani, M. Koddenbrock, M. Palmieri, editors, Software Engineering and Formal Methods. SEFM 2022 Collocated Workshops, 5–16, Springer International Publishing, Cham, 2023, doi:10.1007/978-3-031-26236-4_1.

[7] E. C. Pinto Neto, D. M. Baum, J. R. d. Almeida, J. B. Camargo, P. S. Cugnasca, "Deep Learning in Air Traffic Management (ATM): A Survey on Applications, Opportunities, and Open Challenges," Aerospace, **10**(4), 2023, doi:10.3390/aerospace10040358.

[8] J. C.-W. Lin, G. Srivastava, Y.-D. Zhang, "Special Issue Editorial: Advances in Computational Intelligence for Perception and Decision-Making for Autonomous Systems," ISA Transactions, **132**, 1–4, 2023, doi:10.1016/j.isatra.2023.01.031.

[9] Y. Ijiri, M. Sakuragi, Shihong Lao, "Security Management for Mobile Devices by Face Recognition," in 7th International Conference on Mobile Data Management (MDM'06), 49–49, IEEE, 2006, doi:10.1109/MDM.2006.138.

[10] A. I. Awad, A. Babu, E. Barka, K. Shuaib, "AI-powered biometrics for Internet of Things security: A review and future vision," Journal of Information Security and Applications, **82**, 103748, 2024, doi:https://doi.org/10.1016/j.jisa.2024.103748.

[11] H. Cui, Z. Chen, Y. Xi, H. Chen, J. Hao, "IoT Data Management and Lineage Traceability: A Blockchain-based Solution," in 2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops), 239–244, IEEE, 2019, doi:10.1109/ICCChinaW.2019.8849969.

[12] N. M. Gonzalez, W. A. Goya, R. de Fatima Pereira, K. Langona, E. A. Silva, T. C. Melo de Brito Carvalho, C. C. Miers, J.-E. Mangs, A. Sefidcon, "Fog computing: Data analytics and cloud distributed processing on the network edges," in 2016 35th International Conference of the Chilean Computer Science Society (SCCC), 1–9, IEEE, 2016, doi:10.1109/SCCC.2016.7836028.

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with deep convolutional neural networks," Communications of the ACM, **60**(6), 84–90, 2017, doi:10.1145/3065386.

[14] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," CoRR, **abs/1409.1556**, 2014, doi:10.48550/arXiv.1409.1556.

[15] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778, IEEE, 2016, doi:10.1109/CVPR.2016.90.

[16] Y. Huang, H. Hu, C. Chen, "Robustness of on-Device Models: Adversarial Attack to Deep Learning Models on Android Apps," in 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 101–110, IEEE, 2021, doi:10.1109/ICSE-SEIP52600.2021.00019.

[17] S. Han, J. Pool, J. Tran, W. J. Dally, "Learning both Weights and Connections for Efficient Neural Networks," Advances in neural information processing systems, **28**, 2015, doi:10.48550/arXiv.1506.02626.

[18] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," CoRR, **abs/1806.08342**, 2018, doi:10.48550/arXiv.1806.08342.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," CoRR, **abs/1704.04861**, 2017, doi:10.48550/arXiv.1704.04861.

[20] Y. Guo, "A Survey on Methods and Theories of Quantized Neural Networks," CoRR, **abs/1808.04752**, 2018, doi:10.48550/arXiv.1808.04752.

[21] Y. Zhao, I. Shumailov, R. Mullins, R. Anderson, "To compress or not to compress: Understanding the Interactions between Adversarial Attacks and Neural Network Compression," Proceedings of Machine Learning and Systems, **1**, 230–240, 2020, doi:10.48550/arXiv.1810.00208.

[22] R. Bernhard, P.-A. Moellic, J.-M. Dutertre, "Impact of Low-Bitwidth Quantization on the Adversarial Robustness for Embedded Neural Networks," in 2019 International Conference on Cyberworlds (CW), 308–315, IEEE, 2019, doi:10.1109/CW.2019.00057.

[23] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, "The Space of Transferable Adversarial Examples," arXiv preprint arXiv:1704.03453, 2017, doi:10.48550/arXiv.1704.03453.

[24] L. Wu, Z. Zhu, C. Tai, W. E, "Understanding and Enhancing the Transferability of Adversarial Examples," arXiv preprint arXiv:1802.09707, 2018, doi:10.48550/arXiv.1802.09707.

[25] A. Kurakin, I. Goodfellow, S. Bengio, "Adversarial examples in the physical world," CoRR, **abs/1607.02533**, 2017, doi:10.48550/arXiv.1607.02533.

[26] W. Brendel, J. Rauber, M. Bethge, "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models," arXiv preprint arXiv:1712.04248, 2018, doi:10.48550/arXiv.1712.04248.

[27] N. Carlini, D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in 2017 IEEE Symposium on Security and Privacy (SP), 39–57, IEEE, 2017, doi:10.1109/SP.2017.49.

[28] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, X. Yi, "A Survey of Safety and Trustworthiness of Deep Neural Networks: Verification, Testing, Adversarial Attack and Defence, and Interpretability," Computer Science Review, **37**, 100270, 2020, doi:10.48550/arXiv.1812.08342.

[29] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, "The Limitations of Deep Learning in Adversarial Settings," CoRR, **abs/1511.07528**, 2015, doi:10.48550/arXiv.1511.07528.

[30] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, F. Roli, "Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks," in 28th USENIX security symposium (USENIX security 19), 19, 2019, doi:10.48550/arXiv.1809.02861.

[31] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 60, 2009.

[32] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, **86**(11), 2278–2324, 1998, doi:10.1109/5.726791, conference Name: Proceedings of the IEEE.

[33] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2574–2582, 2016, doi:10.48550/arXiv.1511.04599.

[34] A. G. Matachana, K. T. Co, L. Muñoz-González, D. Martinez, E. C. Lupu, "Robustness and Transferability of Universal Attacks on Compressed Models," CoRR, **abs/2012.06024**, 2020, doi:10.48550/arXiv.2012.06024.

[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," arXiv preprint arXiv:1706.06083, 2019, doi:10.48550/arXiv.1706.06083.

[36] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, "Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks," in 2016 IEEE symposium on security and privacy (SP), 582–597, IEEE, 2016, doi:10.48550/arXiv.1511.04508.

[37] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, Y. Zou, "DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients," CoRR, **abs/1606.06160**, 2018, doi:10.48550/arXiv.1606.06160.

[38] Y. Bengio, N. Léonard, A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," CoRR, **abs/1308.3432**, 2013, doi:10.48550/arXiv.1308.3432.

[39] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks," **9908**, 525–542, 2016, doi:10.1007/978-3-319-46493-0_32, series Title: Lecture Notes in Computer Science.

[40] M. Courbariaux, Y. Bengio, J.-P. David, "BinaryConnect: Training Deep Neural Networks with binary weights during propagations," Advances in neural information processing systems, **28**, 9, 2015, doi:10.48550/arXiv.1511.00363.

[41] TensorFlow Lite, "TensorFlow Lite | ML for Mobile and Edge Devices," 2021.

[42] Y. Zhao, X. Gao, R. Mullins, C. Xu, "Mayo: A Framework for Auto-generating Hardware Friendly Deep Neural Networks," in Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning, 25–30, ACM, 2018, doi:10.1145/3212725.3212726.

[43] M. Jaderberg, A. Vedaldi, A. Zisserman, "Speeding up Convolutional Neural Networks with Low Rank Expansions," CoRR, **abs/1405.3866**, 2014, doi:10.48550/arXiv.1405.3866.

[44] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, "Adversarial Examples Are Not Bugs, They Are Features," Advances in neural information processing systems, **32**, 2019, doi:10.48550/arXiv.1905.02175.

[45] Y. Liu, X. Chen, C. Liu, D. Song, "Delving into Transferable Adversarial Examples and Black-box Attacks," CoRR, **abs/1611.02770**, 2017, doi:10.48550/arXiv.1611.02770.

[46] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1–9, IEEE, 2015, doi:10.1109/CVPR.2015.7298594.

[47] K. Duncan, E. Komendantskaya, R. Stewart, M. Lones, "Relative Robustness of Quantized Neural Networks Against Adversarial Attacks," in 2020 International Joint Conference on Neural Networks (IJCNN), 1–8, IEEE, 2020, doi:10.1109/IJCNN48605.2020.9207596.

[48] X. Huang, M. Kwiatkowska, S. Wang, M. Wu, "Safety Verification of Deep Neural Networks," **10426**, 3–29, 2017, doi:10.1007/978-3-319-63387-9_1, series Title: Lecture Notes in Computer Science.

[49] J. Rauber, W. Brendel, M. Bethge, "Foolbox: A Python toolbox to benchmark the robustness of machine learning models," CoRR, **abs/1707.04131**, 2018, doi:10.48550/arXiv.1707.04131.

[50] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, R. Long, P. McDaniel, "Technical Report on the CleverHans v2.1.0 Adversarial Examples Library," CoRR, **abs/1610.00768**, 2018, doi:10.48550/arXiv.1610.00768.

[51] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy, B. Edwards, "Adversarial Robustness Toolbox v1.0.0," CoRR, **abs/1707.04131**, 2019, doi:10.48550/arXiv.1807.01069.

[52] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," CoRR, **abs/1602.02830**, 2016, doi:10.48550/arXiv.1602.02830.

[53] J. Uesato, B. O'Donoghue, A. v. d. Oord, P. Kohli, "Adversarial Risk and the Dangers of Evaluating Against Weak Attacks," in International conference on machine learning, 5025–5034, PMLR, 2018, doi:10.48550/arXiv.1802.05666.

[54] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, C.-J. Hsieh, "ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models," 15–26, 2017, doi:10.1145/3128572.3140448.

[55] Y. Li, X. Dong, W. Wang, "Additive Powers-of-Two Quantization: An Efficient Non-uniform Discretization for Neural Networks," CoRR, **abs/1909.13144**, 2020, doi:10.48550/arXiv.1909.13144.

[56] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning," in NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, 9, 2011.

[57] A. Shrestha, J. Großmann, "Adversarial Neural Network Toolkit," https://github.com/Abhishek2271/AdversarialNeuralNetworkToolkit, 2024.

[58] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, "TensorFlow: A system for large-scale machine learning," OSDI'16, 21, USENIX Association, 2016, doi:10.48550/arXiv.1605.08695.

[59] Y. Wu, et al., "Tensorpack," https://github.com/tensorpack/, 2016.