Highlights

Unsupervised learning models-based CRM anomaly detection using GPU

Daniel Bastidas,Olmer Garcia-Bedoya,Oscar M. Granados

- We implemented a framework of CRM anomaly detection on GPU.
- The approach focused on the application of deep learning-based anomalies detection algorithms.
- The model could be easily used in other deep learning methods to improve the computing cost in a CRM platform.

Unsupervised learning models-based CRM anomaly detection using GPU

Daniel Bastidas[a], Olmer Garcia-Bedoya[a] and Oscar M. Granados[b,*]

[a]Department of Industries and Digital Technologies, Universidad Jorge Tadeo Lozano, Bogotá, Colombia
[b]Department of Economics and International Trade, Universidad Jorge Tadeo Lozano, Bogotá, Colombia

ARTICLE INFO

Keywords:
Anomaly detection
Customer relationship management (CRM)
Cuda Core
Deep Learning
Graphics processing unit (GPU)

ABSTRACT

Deep learning models have improved several business intelligence tools like Customer relationship Management(CRM) systems. However, those models have increased the need for advanced computational capacity and infrastructure. Modern accelerators are starting to have floating-point precision arithmetic problems generated by highly streamlined systems, powered by the need to process an ever-increasing volume of data and increasingly complex models to attend to the necessity to identify customer data that allow consolidating products or services. We focus on CRM anomalies detection using GPU(Graphics Processor Unit) because they are a relevant source of money drain for organizations and directly affect the relationship between clients and suppliers. Our results present the combination of deep learning models with a computational structure that could access by organizations, but with a combination that reduces the number of features that achieve answers to CRM system.

1. Introduction

The term customer relationship management (CRM) has been used interchangeably. Several scholars and executives have used it to reflect a variety of themes and perspectives that connect customers with the products and services of companies [20, 24]. A CRM is used by almost every company to handle their daily sales operations. CRM is a sophisticated storehouse of massive amounts of data that is expanding and open to irregularities. Every organization uses reports to evaluate, assess, or improve the performance of products, services, campaigns, advertisements, and people. Therefore, the lack of a specific capability to help organizations deal with anomalies is one of the primary challenges that several corporations with obsolete CRMs confront. Manually identifying anomalies might cost hours of work and put us behind in the battle to manage them.

Powered by the inexhaustible need to process an ever-increasing volume of data and increasingly complex models, modern accelerators are starting to support floating-point precision arithmetic in specialized hardware. Highly streamlined systems such as Tensor Cores and CUDA Cores on NVIDIA GPUs and Tensor Processing Units (TPU) developed by Google that provide matrix multiplication of exceptionally efficient precision [32], where next-gen Nvidia graphics cards can get to process up to 30 times faster than last generation processors [19]. In this article, we use GPUs principally for the training of unsupervised models for multiple-task learning, aimed at the detection of CRM anomalies, which are demanding in terms of the level of processing and, if possible direct these efforts to make the churn rate even lower by integrating the Customer Relationship Management (CRM) with our new model.

The main issue with unsupervised learning methods lies in their potential to use large amounts of unstructured data and, in turn, learn complex and highly non-linear models with millions of free parameters [16]. Unfortunately, current learning algorithms for unsupervised methods are too slow for large-scale applications, forcing researchers to focus on smaller-scale models or use fewer training examples. For this, massively parallel methods are suggested to help solve these problems, arguing that modern graphics processors far exceed the computational capabilities of multi-core CPUs(Central Processing Unit) and have the potential to revolutionize the applicability of specific unsupervised learning methods.

If we introduce deep-learning models to the formula using massive data, the task becomes increasingly difficult to process, especially when training these kinds of models. So it is relevant to investigate this field of application and help task automation with these new findings and recent technologies such as the Nvidia Tensor Core Units [7]. The purpose of these learning models will be to focus on detecting such CRM anomalies since it is a vital source of money drain for companies and directly affects the relationship between clients and suppliers.

Researchers and developers must learn the complex structure from chaotic data to build an efficient anomaly detection system, identifying dynamic anomaly patterns and detecting out liners without having enough data [30, 23]. First, we introduce the anomaly detection problem and approaches adopted before the deep model era in conjunction with the challenges we face. Then, we will examine some state-of-the-art deep learning models and compare the techniques used to overcome the limitations of traditional algorithms.

However, a common problem in those studies is false positives and negatives [8] since not all the anomalies that occur are actually of fraudulent origin. We plan to implement expert opinions to improve this information for the al-

*Corresponding author
daniel.bastidasb@utadeo.edu.co (D. Bastidas);
olmer.garciab@utadeo.edu.co (O. Garcia-Bedoya);
oscarm.granadose@utadeo.edu.co (O.M. Granados)
ORCID(s): 0000-0002-6964-3034 (O. Garcia-Bedoya);
0000-0002-4992-8972 (O.M. Granados)

gorithmic adjustments on the solution that to face the following requirements. First, to identify anomalies without damaging the privacy of those who provide their data. Second, to obtain a comprehensive understanding of anomaly detection techniques based on deep learning in several application domains. Third, to make a benchmark-type comparison of performance in different platforms and architectures. Fourth, to study deep model anomaly detection techniques in the real world.

The article aims to give an overview of deep learning models for CRM anomaly detection using GPU. The approaches use a dataset were 22 different SQL Server databases which range from 2008 to 2019 versions. The main contribution is CRM anomalies detection using GPU that combines deep learning models with a computational structure that could access by organizations. To show the behavior of our proposed, we present a combination of Deep Convolutional Neural Networks(CNN) and long short-term memory(LSTM) AutoEncoder that provides a provision to detect anomalies in real-time streaming data designed to run on GPU/CPU.

The outline of the paper is as follows. Section 2 introduces some basic concepts about CRM. Section 3 presents the related works about GPU-based analysis frameworks to meet the demands of growing data. Section 4 gives an overview of materials, especially the data methodology, structure, and analysis. In section 5, we explain the methods, and in the following section, we present the results. Section 7 discusses the results and Section 8 concludes this article and points out new research directions.

2. Preliminaries

We provide some definitions of the fundamental terminology required in our analysis.

CRM. Customer relationship management is an administering approach that seeks to create, develop and enhance relationships [3]. In the digital ecosystem, the CRM is a process in which an organization manages its interactions with customers using different datasets which try to improve the relations with them through data analysis and the study of information that allow defining new strategies [17, 29]. However, the benefits of eCRM and its compatibility depend of the technological and business context of the firm [25, 6].

Additionally, CRM is a complex ever-growing data storehouse that is prone to anomalies frequently. We always come across anomalies if the organization utilizes reports to evaluate and asset the performance of persons, goods, services, campaigns, and advertisements. The lack of a specific capability to help organizations deal with anomalies is one of the primary challenges companies with obsolete CRMs confront. In the CRM, we have three main phases where the client can or cannot proceed with his final order according to a forecast, responsible for this client, commercial conditions, work-group, and many other parameters relevant to our goal. We describe the three main stages:

- *Business Opportunity.* In this phase, we have the first encounter with possible clients, where the commercial employees will create the client with his basic information, the channel of communication, and a percentage of interest to buy. With this information and other fields, the CRM will have a certain forecast assigning a priority to this client in its program interface.
- *Quotation Request.* For this process, we assume that the client already has a concrete list of products that needs, and changes the forecast to a more positive one, also specifying the person responsible for the client, commercial conditions of the purchase, discounts, and other parameters that they define the business specifically.
- *Customer Order.* With products confirmed in the quotation, we proceed to place the customer order, which consolidates the purchase requested by the customer with the agreed price, the delivery date, the guarantee, payment method, and maturity of the order, if it applies to other commercial conditions can be associated as well.

3. Related Work

The need for GPU-based analysis frameworks to meet the demands of growing data in different industries has been acknowledged by scholars. Guo et al. [11] proposed an architecture design and execution model that offers programmability of general use in DNN accelerators to accelerate end-to-end applications (simultaneous multimode architecture - SMA). The key of this architecture is the temporal integration of the systolic execution model with the GPU-type SIMD execution. Additionally, it exploits the components shared between the systolic array throttle and the GPU and provides reconfiguration capability lightweight to switch between the two modes on-site. SMA achieves performance improvement up to 63% while consuming 23% less power than the Volta architecture of baseline with TensorCores. Sadhu et al. [27] suggest semi-supervised anomaly identification using CAN Bus Scalar Sensor Data, which takes into account the imbalance in normal conditions. Driving data contains different positive/normal circumstances (e.g., right turn, straight), some of which (e.g., U-turn) are as uncommon as anomalous situations. Zhang and Wu [33] present a large-scale mixed-precision linear least-square solver that uses the low-precision TensorCore GPU to achieve high accuracy. The mixed precision method is up to 14 times faster than single-precision cuSOLVER at QR matrix factorization on a broad scale with marginally lower accuracy and up to 10 times faster than double precision direct QR least-square solver, with comparable accuracy. While Dakkak et al. [7] propose an algorithm that uses idle NVIDIA TCUs to achieve 89% to 98% peak memory copy bandwidth while being orders of magnitude faster (up to 100x for reduction and 3x for the scan) than state-of-the-art methods for small segment sizes, which are popular in machine learning and science applications. The algorithm accomplishes this, thus reducing power consumption by up to 22% for reduction and 16% for scanning.

Another group of scholars is concentrated in the Multi-Task context. First, Liu et al. [15] developed a multi DNN for learning representations through various tasks that take advantage of vast volumes of cross-task data while also taking advantage of a regularization effect that contributes to representations that can aid tasks in new domains. Their multi-task DNN methodology incorporates multiple-domain classification (for query classification) and information retrieval (ranking for web search) and reveals advances over high baselines in a wide variety of domain adaptation tasks. The approach of Akhtar et al. [1] takes advantage of the relatedness among the participating activities, defining three multi-task configurations, each with two tasks: sentiment eight emotion classification, sentiment classification 8 sentiment intensity prediction, and emotion classification 8 emotion intensity prediction. The proposed method outperforms the single-task learning framework on the CMU-Multi-modal Opinion Sentiment and Emotion Intensity benchmark dataset, indicating that the evolved multi-task framework outperforms the single-task learning framework for the inter-related participating tasks. The proposed multi-task learning system by Huang et al. [12] uses deep neural networks (DNNs) to learn and refine two companion tasks in Web search engines: entity suggestion and document rating. In particular, when it comes to document ranking as a support role for improving the main task of entity recommendation, where the representation of requests, sessions, and users are exchanged across all tasks and optimized during training by the multi-task objective. In [18] it is manifested a difficulty with selecting the network architecture and hyperparameters, which has been solved using current architectural search algorithms that trade the human experience for computation time.

In the context of anomaly detection, Zheng et al. [34] focus on solder paste inspection (SPI), a critical stage in the semiconductor manufacturing process that detects irregular boards. They propose a fast clustering algorithm that avoids retraining and fine-tuning in the inference process by reusing existing models. They assess their strategy using data obtained from production lines over three months, and they can reduce false alarms by 81.28%. that represents a $11.3 million annual savings. Another example is the paper of Das et al., [8], where they define the Active Anomaly Discovery (AAD) algorithm, which integrates expert user input to mark a queried data instance as an anomaly or nominal point. This input is used to change the anomaly detector so that the outliers it detects are more in line with the semantic interpretation of the anomalies by the expert consumer. The AAD algorithm uses a weighted ensemble of anomaly detectors to identify them. When it receives a mark from the consumer, it changes the heights of individual ensemble members such that anomalies rank higher than outliers in terms of anomaly score.

Coussement et al. [5] compare an integrated logit model to eight state-of-the-art data mining techniques that use common input data, including real-world cross-sectional data from a European telecommunication provider. In the sense of churn prediction modeling, they noticed improvements of up to 14.5% in the region under the receiving operating characteristics curve and 34% in the top decile lift when analysts realize that the data-preparation technique they choose affects churn prediction results. Huang & Kechadi [13]: defined that using a single model-based classification system to produce a satisfactory outcome could not be enough. They propose a novel hybrid model-based learning method that combines supervised and unsupervised approaches for forecasting consumer actions to achieve better predictive results. An updated k-means clustering algorithm and a classic rule inductive technique are used throughout the method (FOIL). Lindemann et al. [14] compared the different anomaly detection systems, emphasizing the types of anomalies that exist, data types, architectures, applicability, scenarios, metrics, and performance. However, they defined that the solutions proposed with graphs using clustering and scores by cluster are for certain types of problems relevant because currently, there are no implementations of LSTM networks into graph-based approaches for characterization of contextual anomalies. Finally, Chen et al. [4] propose a method for predicting consumer turnover directly from experimental behavioral evidence using ensemble techniques. The hierarchical multiple kernel support vector machine (HMK-SVM) is the solution proposed. The H-MK-SVM is trained using a three-phase training algorithm that has been designed, applied, and validated. During the training period, the H-MK-SVM creates a classification feature by coefficients calculation of static and longitudinal behavioral variables without transforming the longitudinal behavioral data. All of this can be compared by library usage and hardware availability on Table 1.

4. Materials

4.1. Data Methodology

The paper implemented CRISP-DM methodology[31] which views the data analysis process as a professional project, thus establishing a context that influences the modeling. This context takes into account the existence of a client that is not part of the development team, as well as the fact that the project not only does not end once the model is found (since it requires deployment and maintenance afterward) rather, it can be related to other projects. Hence it needs to be fully documented for other development teams to use and build on the knowledge. The life cycle of a data mining project consists of six phases that are not strictly rigid, i.e., moving back and forth between them is allowed. Each result determines which phase or what particular task needs to do next. This methodology accepts the moving back and forth between different phases.

The methodology identifies the cycle as a business understanding (Defining customer needs), data understanding, data preparation, the modeling definition, the model evaluation, especially with the business objectives. Even if the model objective is to increase knowledge of data, the knowledge obtained will have to be organized and presented to use

Article	Algorithms	Libraries	Hardware
[11]	Deep neural networks (DNN) Simultaneous Multi-mode Architecture (SMA) Non-max suppression	CUTLASS	NVidia GPUs
[33]	QR factorization LLS High Accuracy Solver Direct Linear Least Square Problem Solver Recursive Modified Gram-Schmidt Recursive tiled HOUseholder	CUTLASS WMMA cuBLAS cuSOLVER	NVidia V100 Google TPUs
[7]	General matrix-matrix multiplication (GEMM) General matrix-vector multiplication (GEMV) Reduction256N Scan256N	CUBLAS CUTLASS WMMA API	NVIDIA V100 Cuda Cores
[26]	Deep Belief Networks (DBN) Sparse coding Parallel L1-regularized least squares	ATLAS BLAS Goto BLAS	Nvidia GeForce GTX 280 Dual-core CPU @ 3.16GH
[15]	Multi-task DNN (Deep Neural Network) Stochastic gradient descent (SGD) Support Vector Machine (SVM) SemanticRepresentation (MT-DNN)	Word2Vec Word3gram Letter3gram	Multi-Core CPU
[1]	Multi-task Multi-modal Emotion and Sentiment (MMES) Multi-task learning (MTL) framework	Keras Tensorflow	Google TPUs
[12]	Multi-task DNN (Deep Neural Network) BiLSTM (Bidirectional Long-Short Term Memory) ER-C-MT Gradient Boosted Decision Tree (GBDT) Memory Based Approach (MBR) LTR (context-insensitive model) LTR-ER-C-MT (context-aware model)	Unmentioned	Unmentioned
[34]	Fast Clustering Algorithm (FCA)	Unmentioned	Unmentioned
[8]	Active Anomaly Discovery (AAD)	Unmentioned	Unmentioned

Table 1
Hardware & Software Comparison with Algorithms

it by the client. Depending on the requirements, the development phase can be simple or complex depending on the data analysis process in the organization.

4.2. Data Collection

Here we explain some relevant information about the dataset, describing the raw data and data mining processes required to obtain our samples. First of all, we define the query which unifies the most relevant tables in the CRM. The three main stages explained in Preliminaries, the creation date of the process, status (if in progress, succeeded or canceled), a forecast from 1 to 100 of interest seen by the vendor, the client database generated code, the person who created the process, the person in charge of the process, observations done by the responsible, the total price for products, a representative market exchange rate for non-local orders, reason of cancellation, and other data for a total of 26 columns with approximately 170.000 unique rows for this experiment. The data sources were 22 different SQL server databases which range from 2008 to 2019 versions, all from productive clients in this CRM. Fortunately, every column is ciphered (integer foreign key values), so no sensitive data is filtered and can be applied to any generalization data from available CRM as long as the inputs are in CSV format. With the help of an open-source procedure called "sp-ineachdb" by Brent Oztar [22], it was possible to execute and to insert into a temporal table a globalized query execution in every database available on one specific server.

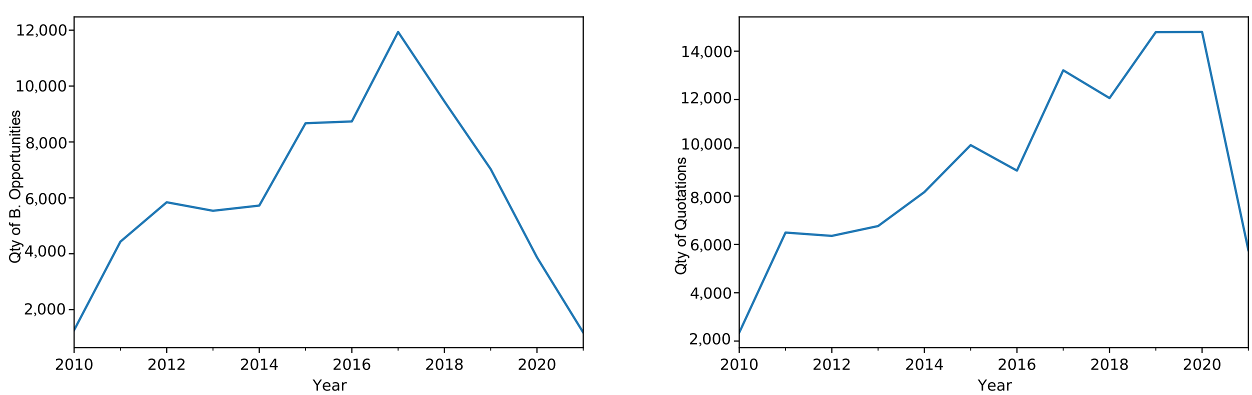
4.3. Data Analysis

In the analysis projected to our data set, we found behavior patterns described below. First, in the Figure 1 of quotations per year and business opportunities, we see a low peak in 2020 due to the operation reduction that the pandemic brought. Second, we identify that the business opportunities per year had their maximum level in 2017, where companies managed to consolidate their core business and were in decline until the 2021 first semester. Furthermore, the companies that achieve stability stopped the use of the business opportunities module. These results are supported by the Figure 1b (quotations per year), where they were growing and found an equilibrium in 2019 and a low peak in 2020.

About 75% of the data in the columns QuotationBuyInterest and OpportunityBuyInterest corresponds to a 50% probability of sale and business opportunity. Namely, advisers never have an accurate prediction of the final decision done by the client, and the probability is equal to whether they buy or not buy for the advisor who performs the fore-

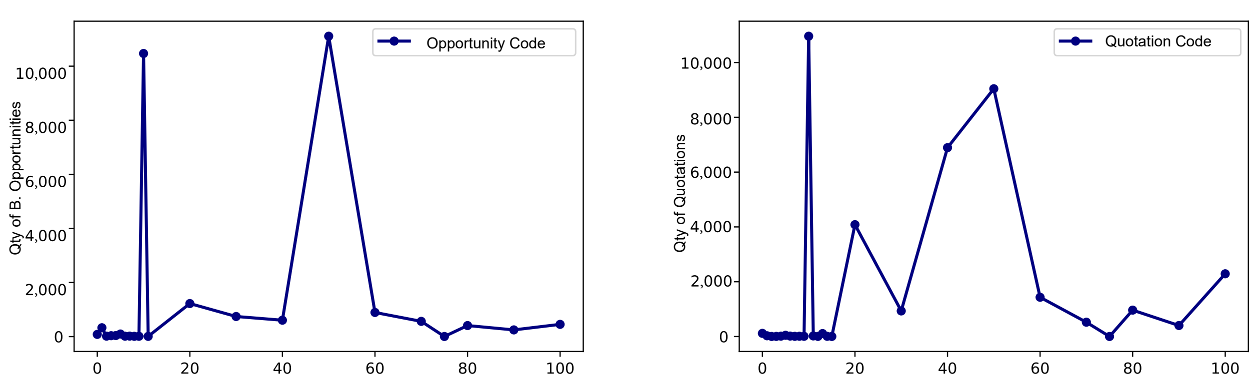
Column Name	DType	Description
ProcessDate	datetime64[ns]	This column is used to define the creation date of any specified process (see chapter 2)
OpportunityBuyInterest	float64	This column is used to define the percentage of interest (noticed by the vendor) in business opportunities
QuotationBuyInterest	float64	This column is used to define the percentage of interest (noticed by the vendor) in quotation requests
QuotationSubtotal	float64	This column unifies the total price for any given process (see chapter 2)
TradingCurrency	float64	This column unifies the trading currency for any given process(see chapter 2)
QuotationCancellation	float64	In this column we save the cancellation reason for a given quotation request
OrderCancellation	float64	In this column we save the cancellation reason for a given customer order
OpportunityCancellation	float64	In this column we save the cancellation reason for a given business opportunity

Table 2
Data Dictionary for CRM (encoded)



(a) (b)

Figure 1: Business Opportunities and Quotations Quantity per Year.



(a) (b)

Figure 2: Business Opportunities and Quotations Quantity Versus % of Interest.

cast measurement.

Another trend is the number of quotations against percentages of interest in which a large number of quotations appear below 50%, adding more than 60 thousand rows below that range, it is not a good quantity considering that the total of quotations is 115 thousand rows, where, almost four thousand rows have 0% interest as shown in figure 2. Thus, those quotations were lost from the moment of creation, it is necessary to find a way to increase the percentage of interest in the forecast per quotations, and for that, we need the anomaly predictor.

In the same graph applied to business opportunities, we see behavior similar to that of quotations in terms of the low forecast where only a 20% probability of purchase predominates as shown in figure 2, it is necessary to be able to adjust a better model according to the business rules. Also, there are external factors such as economic recession, which are out of our control. We are still trying to understand why and how to take back those clients that have been lost between 2017 and 2020, especially during the pandemic period.

For the correlation of variables, we find a strong relationship between the priority and the detail of the cancellation of the quotation, implying that the more priority the quotations have, the less probability of cancellation they have associated, i.e., monitoring is essential to decrease churn rate, as show in figure 3. In addition, there is also a strong relationship between the quotation currency and the reason for quotation cancellation, which also indicates that businesses conducted in foreign currency are often canceled.

5. Methods

5.1. Time Series Data

Time series data refers to information collected over a period. For example, any sensor may be classed as a time series if a collection of sensor data is seen at certain in equal intervals. It is not time-series data when data are collected in any order or all at once. Therefore, detecting suspicious patterns from large data is crucial in any business industry. For example, in the banking industry, fraudulent transactions constitute a severe threat to the bank's loss/liabilities or against money laundering[10].

To evaluate time-series data, we must first understand the pattern types that will be combined to create a collection of observations.

1) Prolonged pattern in the time series is referred to be a trend. It depicts the low, medium, and high-frequency changes that have been filtered out of the time series. Habitually, if there is no growth or decreasing trend in the time series data, it is considered stationary.

Trend patterns have two categories. First, deterministic in nature where the effect of the shocks in the time series are avoided in this situation, and, second, stochastic as a process in which the impacts of shocks are never completely eradicated since the time series' level is altered permanently .

2) Cyclic: The pattern is characterized by up and down movements that revolve around a specific trend. 3) Seasonal: A pattern that shows regular changes. People's seasonal and customary circumstances cause these short-term fluctuations. The data is subjected to regular and predictable modifications that occur at periods (specific and well-defined timeframe). Seasonality comes from a variety of sources, including institutions, climate, social customs, and habits. Models for adding a seasonal component to a time series include:

- Additive Model. The model combination of the seasonal and trend components.
- Multiplicative Model. If there is no trend component in the time series, the seasonal component is multiplied by the intercept in this model.

4) Irregular: It is an unpredictable component of time series. To model successfully the time series is important in machine learning and deep learning. Time series analysis is used to understand the internal structure and functions that are used for producing observations. Time Series analysis is used for:

- Descriptive. Patterns in associated data are recognized. In other words, the time series' changes in trends and seasonality are recognized.
- Explanation. Data is analyzed, and models are created.
- Forecasting. For short-term trends, predictions are made based on prior data.
- Invention Analysis. Any event's effect on time series data is examined.
- Quality Control. It sends out a warning when a certain size deviates.

The model used unsupervised learning to incorporates context, seasonality, and trend into account for detecting anomalies, this was tested using a Deep CNN architecture as proposed by [18] and an autoencoder based in LSTM networks by [28] explained briefly below. It forecasts the next timestamp for a particular time series window, which is then supplied to a detector module, which compares the anticipated value to the actual data point to discover abnormalities in real-time. Even in domains where time-series data is acquired from several sources and sensors, the technique is practical and appropriate. The module recognizes events in multidimensional time series by collecting variations and trends to create a connection. Additionally, to discover associated features and assist in feature selection from raw sensor data.

5.2. Deep CNN

This architecture can identify abnormalities in time-series data, including point anomalies, contextual anomalies, and discords. The methodology does not require labels for anomalies because it is unsupervised. We collect and learn the data

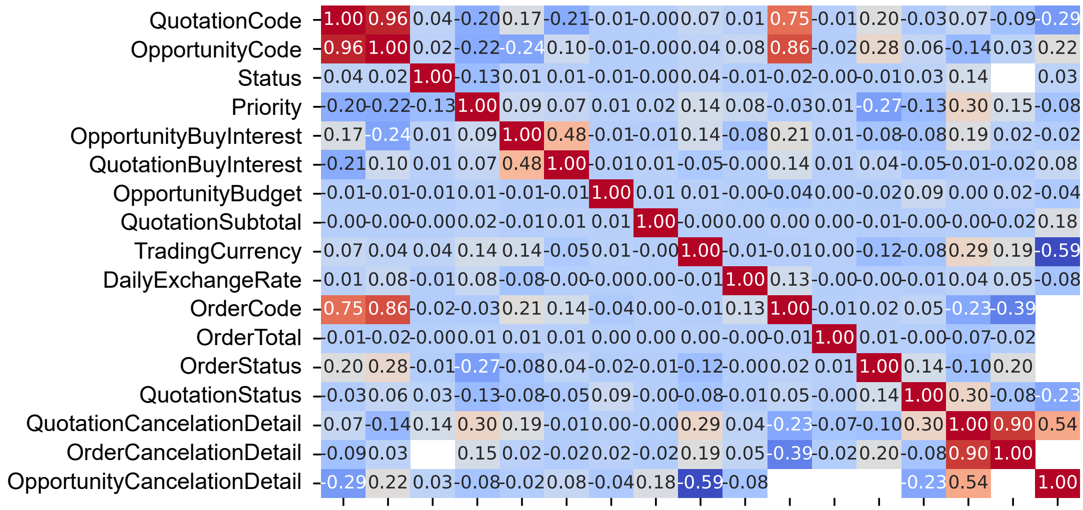


Figure 3: Heatmap of Total Parameters.

distribution utilized to anticipate a time-series usual behavior using unlabeled data. The model design has two parts: a time series predictor and an anomaly detector. First, the time-series predictor employs a deep convolutional neural network to anticipate the next time stamp on the given horizon. This component uses a time series window (as a reference context) to forecast the future timestamp. Second, the anomaly detector component provides the projected value, which determines if the timestamp is non-anomaly or anomaly. The architecture for this model is based in [18].

This design uses two convolutional layers, each followed by a max-pooling layer. We transformed the data into w vectors, i.e., the input layer contains w input nodes. Each layer has 32 filters (kernels) and an elementwise activation function, ReLU. The network's last layer is a fully connected (FC) layer, in which each neuron is linked to all of the neurons in the preceding layer. The network forecast for the next timestamp is represented by this layer. The output layer has the same number of nodes as the input layer, which is p_w [18]. Because we are simply anticipating the next timestamp in our instance, the number of output nodes is only 1. Like other neural networks, the CNN adapts its parameters (weights and biases) to execute the learning task using training data. The network's parameters are tuned using the ADAM optimizer.

5.3. LSTMAENN

The Deep LSTM-based Stacked Autoencoder for Multivariate Time Series is the second architecture. On an unsupervised approach, to learn efficient data representation utilized the autoencoder architecture. The LSTMAENN has several LSTM layers stacked together with input size — the number of anticipated features in the input x, hidden size — the number of features in the hidden status h, and num layers — the number of layers in the hidden status — Recurrent layer count (default: 1), and so on. The LSTMAENN architecture has two LSTM layers, one of which functions as an encoder and the other as a decoder. The encoder layer converts the input sequence $(x_1, x_2, ..., x_n)$ into a representation vector that may be learned. The decoder layer then accepts that vector as input and attempts to reassemble the input sequence $(x_1, x_2, ..., x_n)$. The mean squared error of the difference between the original input sequence and the reconstructed sequence is the cost function for the model [28].

The concept is that an LSTM network contains several gates with training parameters. Some of these gates regulate the module's output, while others regulate forgetting. Because there might be delays of uncertain duration between critical occurrences in a time series, LSTM networks are a suitable fit for categorizing, processing, and generating predictions based on time series data. Furthermore, LSTM comes in handy since it aids in the memory of cell status while retaining data [9, 21].

5.4. Deployment

Three aspects integrated the deployment. First, an open-source low-code Multi-Sensor Data Analysis (MSDA) library that attempts to speed up the hypothesis-to-insights cycle in time-series multi-sensor data analysis and experimentation. It allows to swiftly and efficiently do end-to-end proof-of-concept tests. The module recognizes events in multidimensional time series by recording variations and trends to create a connection to discover linked features and assist in feature selection from raw data. Also, it provides a provision to precisely detect the anomalies in real-time streaming data using an unsupervised deep convolutional neural network and an LSTM autoencoders-based detector designed to run on GPU/CPU.

Second, the hyperparameters applied for both models in common are as following:

- Lookback size: set to 10 determines how many historical data points (including the current row) will be used in the model, also known as the time series window size.
- Dimensions: this parameter is setting dynamically, and it refers to the number of features to be processed by the model, and in our case, it's seven parameters, refer to Table 2.
- Kernel Size: Set to 2 for the DeepCNN, is an integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window.

Third, the model may be trained on either a GPU or a CPU. If GPU is available, the compute function will use it. Otherwise, CPU resources will be used. The Google Colab makes use of the NVIDIA TESLA K80, which is the most widely used GPU, and the NVIDIA TESLA V100, which is the first Tensor Core GPU. The number of training epochs is 30 for our model.

6. Results

The Anomaly Score is the percentage of active columns that were incorrectly anticipated. On the other hand, anomaly likelihood is the probability that a particular abnormality score indicates a real anomaly. There will be a natural degree of uncertainty in each dataset, resulting in a certain amount of prediction mistakes. This mistake degree is accounted for by anomaly likelihood. We can not utilize this measure in our circumstances since we do not have the ground truth anomaly label. The find anomaly function is used to discover anomalies by producing a hypothesis and computing losses, which are anomaly confidence ratings for particular timestamps in the data collection.

The next stage is to locate the anomalies when the training is done. Thus, we back to our original question: how can we accurately evaluate and trace what constitutes an anomaly? Anomaly Score, Anomaly Likelihood, and other newer measures, such as Mahalanobis distance-based confidence score [2], can all be used. The Mahalanobis confidence score assumes that pre-trained neural classifiers' intermediate features follow class conditional Gaussian distributions with equal covariances for all distributions. Additionally, the confidence score for a new input is defined as the Mahalanobis distance from the closest class conditional distribution.

When observing this last graph, we realize that there are great possibilities of anomalous observations above the threshold of 0.4, they accumulate indefinitely for several consecutive years, and a midpoint of stability is found at 0.3 for the period 2010 to 2020. However, for the threshold of 0.6, we found some atypical points in early 2016, possibly due to the VAT hike, which went from 16 to 19, but it would be necessary to look in-depth at the cause of this happening. There is another anomalous spike ending in 2019 which can be caused by a price variation or embezzlement. It would be advisable to investigate the specific rows that caused this spike, as seen in Table 4.

SHAP values can show how much each predictor contributes to the target variable (either positively or negatively), and likewise, the SHAP package assigns each variable its own set of SHAP values. In this way, we can explain why each variable receives its prediction and the contributions of the predictor. For our case study, the most influential variables on the model were the Subtotal of the quotation (QuotationSubtotal). This variable is one with the highest variability index and consequently more anomalies. Additionally, it is relevant because it can show any corruption type or internal embezzlement in the company or some effective fraud towards a client. Likewise, the quotation detail of cancellation (QuotationCancelationDetail) presented a high value. This variable is directly related to the type of cancellation that a quotation had at a given moment, which indicates that the business was not successful. Lastly, the currency type (TradingCurrency) variable is positioned as a high-level index of abnormality, possibly, because the US dollars (USD) and Colombian pesos (COP) businesses have large variations with the sale value in the quotations/client orders.

Finally, for our DeepCNN model, observing the confidence versus frequency graph (Figure 4c), we notice a particular behavior in which a distribution frequency maximum is evidenced around 0.20, where this contrasted with a drastic drop in confidence and distribution, i.e., the predictor works under normal conditions, since the higher the anomaly confidence index, the less the frequency of appearance, which is normal since anomalies are by definition much less frequent than the average data of the data set.

Back to the LSTMAENN model, we see considerably correct predictions in the diagram of confidence versus time (Figure 4d), as well as extremely accurate anomaly forecasts, which goes hand in hand with a much longer training period than DeepCNN, which is shorter (484 seconds for LSTM versus 146 seconds for DeepCNN). The curve of original data versus the projected curve is very close, implying greater learning. Nevertheless, the spikes are not the same as those exhibited in DeepCNN. We can observe new points of anoma-

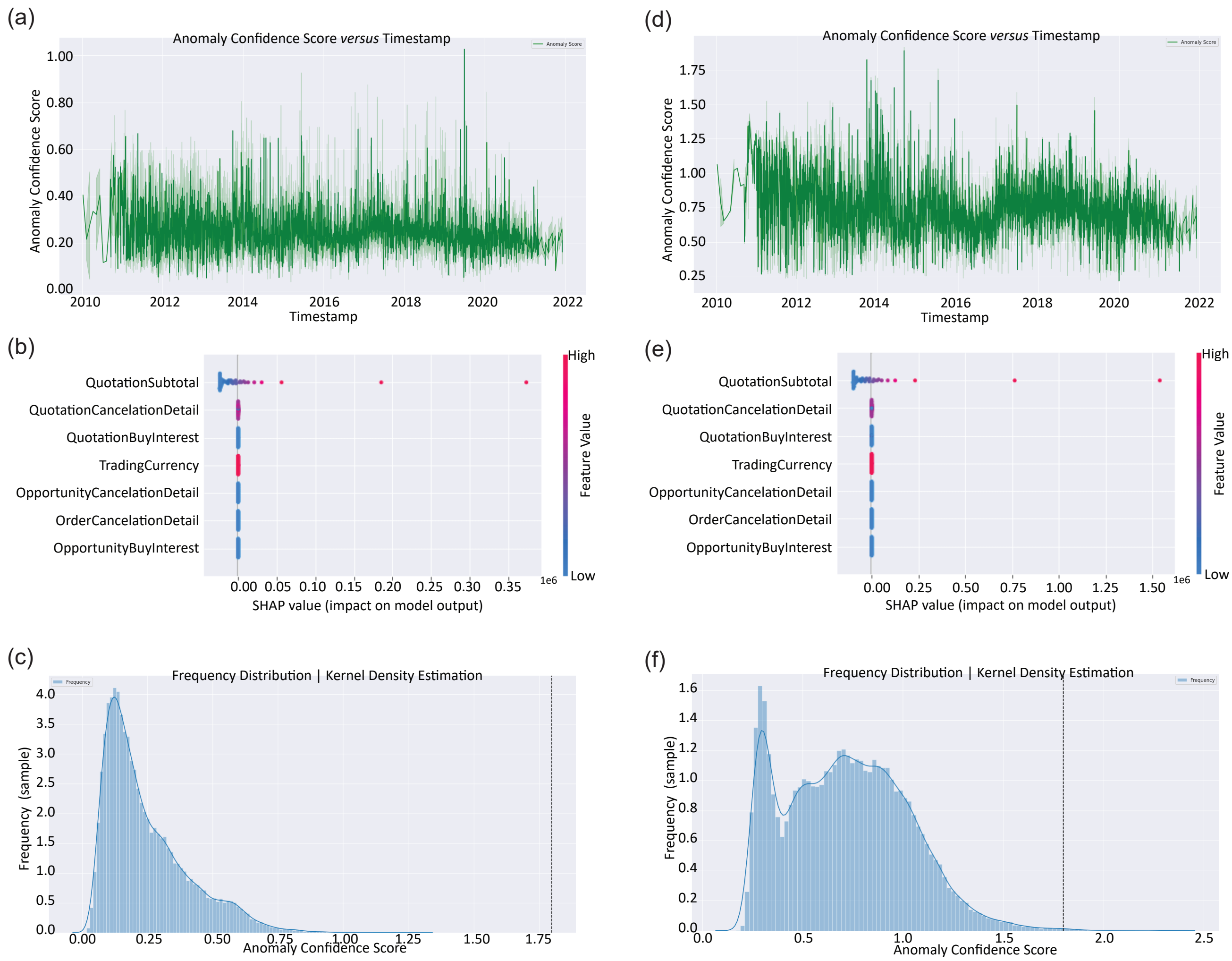


Figure 4: Results DeepCNN and LSTMENN. a. DeepCNN Confidence Vs Time. b. DeepCNN Shap Vs Features. c. DeepCNN Confidence Vs Frequency. d. LSTMAENN Confidence Vs Time. e. LSTMAENN Shap Vs Features. f. LSTMAENN Confidence Vs Frequency.

lies that the previous model failed to identify and is of great importance for the business. From 2014 to 2016, showed marked peaks, some in 2013 and the end of 2017. However, the anomaly scores were higher than those shown in DeepCNN, ranging from 0.25 to 2.0 as maximum (Figure 4a and 4d).

The LSTMAENN confidence graph (Figure 4f) does suffer an impact positively compared to the one shown in DeepCNN (Figure 4c), where we see levels of confidence much lower than the one shown here. Although the frequency is better distributed, the confidence does not drop dramatically, showing better behavior at the training level and accuracy. Only from the point at $X = 1.0$, we see a drop in both frequency and confidence, which is the expected behavior, since when there is not enough data, the anomalies cannot be predicted correctly.

7. Discussion

Anomaly detection is a challenging task among the many possibilities offered by AI. Through observation, analysis, and practice, machine learning is the discipline of teaching a machine to perform a task without the need for human involvement. When enough historical data is provided, training, and regression, we can learn about abnormalities and discover them, saving time, money, and reputation. After experiments, the detection algorithm improves with time, making AI a more efficient alternative to manual detection.

In the variables section, we find some similarities with the one exposed in DeepCNN, except that the variable "QuotationCancelaltionDetail" has been assigned higher priority and with good reason because this variable was foreseen from our prepossessing and analysis phase that it should have had a greater value since it's the one that exposes us to the client if there was a decline on behalf of them to the proposed business. The variable "QuotationSubtotal" still gets the most value on SHAP, as we foresaw in previous sections, due to the importance of this variable in the CRM, we still yet to find how the model uses "TradingCurrency" as it is the currency used in the business process, and can only have two possible values as of now, Colombian pesos or US dollars. During the project development, we have some problems, especially the limitation of the resources to complete the execution of the LSTMAENN model. On several times, we could not finish it due to a lack of VRAM memory. The solution was to reduce the number of columns to process through the MSDA library. Also, we had to contact the library developer due to a bug detected in its source code, introduced in the latest version, and fortunately, it was corrected on time by the developer himself, and we are grateful for that.

8. Conclusion

Learning methods employed during this study showed high effectiveness and high resource consumption at the hardware level (specifically in GPU), where we repeatedly struggled to fit the model into the memory provided by Colab (Tesla T4 of 16GB). However, by optimizing the preprocessing and filtering of data, we were able to fulfill the necessary execution requirements for the CPU processing. Also, we found that it was impossible to complete since the model took hours to execute, and the system ran out of available RAM resources. To better the execution of models with the help of the CUDA and Tensor cores available, GPUs (ideally Nvidia) are becoming more required for autonomous learning in real-world scenarios.

It's fascinating to see how both models behave so differently from one another, given that the output of both models was two-time lines with a score of anomaly probability in each period. The results between the two models were very different, starting with the loss, which was much lower in LSTMAENN and maintained at 0.00812 out of 30 epochs compared to the DeepCNN model, which was higher than the latter with 0.01211, almost double the loss. This is due to a lengthier execution time. DeepCNN takes around two minutes while LSTMAENN takes about five minutes, demonstrating that the work done in LSTMAENN is heavier and optimal in all aspects. The possibility to evaluate the model's performance and robustness using Nvidia Tensorcores is still pending. At this moment, only Cuda cores were utilized by PyTorch.

CRediT authorship contribution statement

Daniel Bastidas: Conceptualization of this study, Data curation, Software. **Olmer Garcia-Bedoya:** Software, Methodology, Writing - Original draft preparation. **Oscar M. Granados:** Data visualization, Methodology, Writing - Original draft preparation.

References

[1] Akhtar, M.S., Chauhan, D.S., Ekbal, A., 2020. A deep multi-task contextual attention framework for multi-modal affect analysis. ACM Trans. Knowl. Discov. Data 14. doi:10.1145/3380744.

[2] Brereton, R.G., Lloyd, G.R., 2016. Re-evaluating the role of the mahalanobis distance measure. Journal of Chemometrics 30, 134–143. doi:https://doi.org/10.1002/cem.2779.

[3] Chen, I., Popovich, K., 2003. Understanding customer relationship management (crm): People, process and technology. Business Process Management Journal 9, 672–688. doi:10.1108/14637150310496758.

[4] Chen, Z.Y., Fan, Z.P., Sun, M., 2012. A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. European Journal of Operational Research 223, 461–472. doi:https://doi.org/10.1016/j.ejor.2012.06.040.

[5] Coussement, K., Lessmann, S., Verstraeten, G., 2017. A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. Decision Support Systems 95, 27–36. doi:https://doi.org/10.1016/j.dss.2016.11.007.

[6] Cruz-Jesus, F., Pinheiro, A., Oliveira, T., 2019. Understanding crm adoption stages: empirical analysis building on the toe framework. Computers in Industry 109, 1–13. doi:https://doi.org/10.1016/j.compind.2019.03.007.

[7] Dakkak, A., Li, C., Xiong, J., Gelado, I., Hwu, W.m., 2019. Accelerating reduction and scan using tensor core units, in: Proceedings of the ACM International Conference on Supercomputing, Association for Computing Machinery, New York, NY, USA. p. 46–57. URL: https://doi.org/10.1145/3330345.3331057, doi:10.1145/3330345.3331057.

[8] Das, S., Wong, W.K., Dietterich, T., Fern, A., Emmott, A., 2020. Discovering anomalies by incorporating feedback from an expert. ACM Trans. Knowl. Discov. Data 14. doi:10.1145/3396608.

[9] Gers, F., Schmidhuber, J., Cummins, F., 2000. Learning to forget: Continual prediction with lstm. Neural Computation 12, 2451–2471. doi:10.1162/089976600300015015.

[10] Guevara, J., Garcia-Bedoya, O., Granados, O., 2020. Machine learning methodologies against money laundering in non-banking correspondents, in: Florez, H., Misra, S. (Eds.), Applied Informatics, Springer International Publishing, Cham. pp. 72–88.

[11] Guo, C., Zhou, Y., Leng, J., Zhu, Y., Du, Z., Chen, Q., Li, C., Yao, B., Guo, M., 2020. Balancing efficiency and flexibility for dnn acceleration via temporal gpu-systolic array integration, in: 2020 57th ACM/IEEE Design Automation Conference (DAC), pp. 1–6. doi:10.1109/DAC18072.2020.9218732.

[12] Huang, J., Wang, H., Zhang, W., Liu, T., 2020. Multi-task learning for entity recommendation and document ranking in web search. ACM Trans. Intell. Syst. Technol. 11. URL: https://doi.org/10.1145/3396501, doi:10.1145/3396501.

[13] Huang, Y., Kechadi, T., 2013. An effective hybrid learning system for telecommunication churn prediction. Expert Systems with Applications 40, 5635–5647. doi:https://doi.org/10.1016/j.eswa.2013.04.020.

[14] Lindemann, B., Maschler, B., Sahlab, N., Weyrich, M., 2021. A survey on anomaly detection for technical systems using lstm networks. Computers in Industry 131, 103498. doi:https://doi.org/10.1016/j.compind.2021.103498.

[15] Liu, X., Gao, J., He, X., Deng, L., Duh, K., Wang, Y.Y., 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval, in: Human Language Technologies-NAACL, The Association for Computational Linguistics. p. 912–921.

[16] Lydia, E.L., Kannan, S., SumanRajest, S., Satyanarayana, S., 2020. Correlative study and analysis for hidden patterns in text analytics unstructured data using supervised and unsupervised learning techniques. International Journal of Cloud Computing 9, 150–162. doi:10.1504/IJCC.2020.109373.

[17] Malthouse, E., Haenlein, M., Skiera, B., Wege, E., Zhang, M., 2013. Managing customer relationships in the social media era: Introducing the social crm house. Journal of Interactive Marketing 27, 270–280. doi:10.1016/j.intmar.2013.09.008.

[18] Munir, M., Siddiqui, S.A., Dengel, A., Ahmed, S., 2019. Deepant: A deep learning approach for unsupervised anomaly detection in time

series. IEEE Access 7, 1991–2005. doi:10.1109/ACCESS.2018.2886457.

[19] Navarro, C.A., Carrasco, R., Barrientos, R.J., Riquelme, J.A., Vega, R., 2021. Gpu tensor cores for fast arithmetic reductions. IEEE Transactions on Parallel and Distributed Systems 32, 72–84. doi:10.1109/TPDS.2020.3011893.

[20] Nevin, J., 1995. Relationship marketing and distribution channels: Exploring fundamental issues. Journal of the Academy of Marketing Science 23, 327–334. doi:10.1177/009207039502300413.

[21] Ordóñez, F., Roggen, D., 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors (Switzerland) 16. doi:10.3390/s16010115.

[22] Ozar, B., . Sql-server-first-responder-kit. https://github.com/BrentOzarULTD/SQL-Server-First-Responder-Kit/blob/dev/sp_ineachdb.sql.

[23] Pang, G., Shen, C., Cao, L., Hengel, A.V.D., 2021. Deep learning for anomaly detection: A review. ACM Comput. Surv. 54. doi:10.1145/3439950.

[24] Payne, A., Frow, P., 2005. A strategic framework for customer relationship management. Journal of Marketing 69, 167–176. doi:10.1509/jmkg.2005.69.4.167.

[25] Racherla, P., Hu, C., 2008. ecrm system adoption by hospitality organizations: A technology-organization-environment (toe) framework. Journal of Hospitality & Leisure Marketing 17, 30–58. URL: https://doi.org/10.1080/10507050801978372, doi:10.1080/10507050801978372.

[26] Raina, R., Madhavan, A., Ng, A., 2009. Large-scale deep unsupervised learning using graphics processors, in: Proceedings of the 26th International Conference On Machine Learning, ICML 2009, p. 110. doi:10.1145/1553374.1553486.

[27] Sadhu, V., Misu, T., Pompili, D., 2019. Deep multi-task learning for anomalous driving detection using can bus scalar sensor data, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2038–2043. doi:10.1109/IROS40897.2019.8967753.

[28] Sagheer, A., Kotb, M., 2019. Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. Scientific Reports 9, 19038. URL: https://doi.org/10.1038/s41598-019-55320-6, doi:10.1038/s41598-019-55320-6.

[29] Trainor, K., Andzulis, J., Rapp, A., Agnihotri, R., 2014. Social media technology usage and customer relationship performance: A capabilities-based examination of social crm. Journal of Business Research 67, 1201–1208. doi:10.1016/j.jbusres.2013.05.002.

[30] Wang, R., Nie, K., Wang, T., Yang, Y., Long, B., 2020. Deep learning for anomaly detection, in: Proceedings of the 13th International Conference on Web Search and Data Mining, p. 894–896. doi:10.1145/3336191.3371876.

[31] Wirth, R., Hipp, J., 2000. Crisp-dm: Towards a standard process model for data mining, in: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, Springer-Verlag London, UK. pp. 29–39.

[32] Zachariadis, O., Satpute, N., Gómez-Luna, J., Olivares, J., 2020. Accelerating sparse matrix–matrix multiplication with gpu tensor cores. Computers & Electrical Engineering 88, 106848. doi:https://doi.org/10.1016/j.compeleceng.2020.106848.

[33] Zhang, S., Wu, P., 2019. High accuracy low precision qr factorization and least square solver on gpu with tensorcore. ArXiv abs/1912.05508.

[34] Zheng, Z., Pu, J., Liu, L., Wang, D., Mei, X., Zhang, S., Dai, Q., 2020. Contextual anomaly detection in solder paste inspection with multi-task learning. ACM Trans. Intell. Syst. Technol. 11. URL: https://doi.org/10.1145/3383261, doi:10.1145/3383261.