IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 22, 2020, accepted July 14, 2020, date of publication August 10, 2020, date of current version September 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015616

Intelligent Performance-Aware Adaptation of Control Policies for Optimizing Banking Teller Process Using Machine Learning

BILAL TAY AND AZZAM MOURAD, (Senior Member, IEEE)
Department of Computer Science and Mathematics, Lebanese American University, Beirut 1102 2801, Lebanon
Corresponding author: Azzam Mourad (azzam.mourad@lau.edu.lb)

This work was supported by the Lebanese American University (LAU).

ABSTRACT In the current banking systems and business processes, the permission granted to employees is controlled and managed by the configured access control methods, in which static role-based models focus on access to information and functions. The deployed configuration is not reviewed/updated systematically and is handled manually by managers. Consequently, banks and companies are looking for systems and applications to automate and optimize their business processes and data management intelligently. In this context, the notion of integrating machine learning (ML) techniques in banking business processes has emerged. In order to build an intelligent and systematic solution, we combine in this paper ML and dynamic authorization techniques to enable performance-based policy evaluation into the banking teller process, where policies adapt to the changes recognized by the ML model. The objective of this work is to focus on the banking teller process that may be generalized to other operational banking processes. In this context, we propose in this paper a new model providing Intelligent Performance-Aware Adaptation of Roles and Policy Control using a support vector machine (SVM). We demonstrate that our model is capable of assessing the deployed control policies and updating them systematically with new roles and authorization levels based on tellers' performance, work history, and system constraints. We evaluated different machine learning models on a real dataset generated from a real-life banking environment. Experimental results explore the relevance and efficiency of our proposed scheme in terms of prediction accuracy, required authorizations, transaction time, and employees' working hours.

INDEX TERMS Optimization of tellers business process, banking business process, behavior-aware access control, performance-aware access control, machine learning, SVM, dynamic policy adaptation.

I. INTRODUCTION

At each branch and internal departments in a bank environment, employees are categorized by their job titles, i.e., junior teller, senior teller, supervisor, and manager. Four eyes principal [1] is used where a maker performs a process or a transaction, and a checker reviews it to authorize or reject it. This method helps to avoid unintentional or intentional mistakes or malicious activities that may be performed by an intruder at a branch or internal departments such as customer accounts and credit department. Such errors, mistakes, and malicious activities have dangerous negative impacts, which may cause loss to the bank and customers in terms of money, time, trust, availability, quality of service, and reputation. In this context, some actions need to be approved by two individuals before they can be taken. It is also called the two-man rule or the two-person rule [1]. In this work, we focus on the banking teller process, which may be generalized to other banking business processes. In this context, an employee assigned a role like a junior teller or senior teller processes financial transactions such as debit, credit, transfer, loans, and deals. Each role has an assigned transaction limit. If the transaction processed by a teller exceeds the limit set on his/her role, an authorization request is sent to his/her supervisor who can accept or reject the transaction. Approve means the transaction can be processed against the customer account on the system. Reject means the transaction should be sent back to a teller for modification and correction. The higher role

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

has a more upper limit set. In every bank branch, a group of tellers is responsible for thousands of transactions per day. Tellers are assigned limits on their privileges based on a static access control method to process transactions depending on their history of performance. This performance is composed of the successful transactions being processed along with the errors they make, which are also associated with complexity levels for a fair assessment of the teller performance. Because of the limits assigned to every teller, some transactions should be approved by supervisors before processing, which causes delays in processing customers' demands. Updating the tellers role by assigning new limits to their privileges helps in minimizing the load on supervisors and leads to faster processing of customers' demands. However, this process usually happens manually and requires human efforts and time to evaluate. Automating such a process for improving the tellers' roles by assigning new limits based on their behavior and performance is necessary, which constitutes the primary motivation behind this paper.

In this context, many solutions were proposed in the literature to embed the aforementioned role-based hierarchy and manage the access control in the banking business process such as local access control file [2], discretionary and mandatory access control models (DAC and MAC) [3], generalized role-based access control model (GRBAC) [4], context-role RBAC model [5], temporal RBAC [6], geo-RBAC model [7], and behavior-based access control [8]. However, most of them are static and based on pre-configured systems. Only a few approaches are dynamic and configurable but do not consider the user performance history to review and update access control policies and procedures systematically. In practice, some employees assigned to the same role as junior teller have different performance levels. Some of them make more errors, and others make less, and some are fast, whereas others are slow. Regardless of their performance, the current approaches provide all tellers with the same restrictions as the authorization limit. To illustrate more on the problem through real life scenarios, bank branches process a huge number of transactions on a daily basis of different amounts, starting from minimal amounts to millions of dollars per transaction. As a result, the supervisor is overloaded by authorization requests to be reviewed in order to ensure appropriate decisions, especially during peak hours. The described scenario causes less customer satisfaction and longer waiting time, low team performance, higher transaction processing average time, more stress on tellers and supervisors to finish their tasks on time and delay starting after branch closure.

In order to overcome the aforementioned problems, many approaches have been proposed in the literature to optimize the business processes using predictive machine learning techniques in order to minimize run time overhead. Some of the proposed optimization approaches are based on ordering cases while taking into consideration the rejection rate and mean effort parameters [9], [10]. Others have proposed prediction monitoring techniques that predict the rejection status of each case based on effort, time, and risk using machine learning methods at run time [11], [12]. However, to the best of our knowledge, none of them have considered the user/employee performance history and have systematically reviewed and updated the control policies and procedures accordingly for banking daily processes and operations.

In this paper, we address the existing limitations by enabling dynamic policy adaption into the banking tellers business process based on tellers' activities by proposing a performance-based authorization scheme. Our approach combines policy authorization with machine learning towards achieving (1) systematic assessment and prediction of teller's performance, and (2) dynamic and behavior-aware adaptation of the authorization policy. Our intelligent performance-aware adaptation of roles and policy control allows us to review and reconfigure the deployed access control models based on user activity history and levels of occurred errors. Given the performed user permissions and limits set by role, the user performance over time (e.g., month and year) are analyzed. In this regard, the proposed machine learning model predicts new roles for tellers with new constraints based on activity recognition techniques for analyzing, classifying, and predicting human actions and performance [13], [14]. Consequently, the control policies and teller role will be upgraded or downgraded according to the aforementioned analysis and model decision. In other words, tellers with better performance will be assigned to an existing or new role with higher privileges, and tellers with poor performance will be assigned to a new or existing role with lower privileges (e.g., transaction limit). In summary, the main contributions of this paper are:

- Elaborating a new scheme for optimizing the banking teller process through a dynamic and behavior-aware adaptation model for role management and policy control based on employees' performance. Our approach considers the context and behavior factors related to the tellers' work experience and performance. In this context, the proposed scheme is capable of assessing the deployed control policies and updating them systematically based on the prediction of the behavior analysis module. In this work, we focus on the banking teller process, which may be generalized to other related banking processes.
- Proposing the use of Support Vector Machines as the ML classifier for tellers' behavior and performance prediction. Experiments on real-life data collected and analyzed from several banks' environments explore the relevance and accuracy of our model compared to Multinomial Logistic Regression and Random Forest.
- Evaluating tellers' performance systematically and reducing transactions' and required authorizations' time. This also leads to an improvement in users' satisfaction by serving them faster. Experiments on real-life data also explore that customers are served with less time and illustrate the gain in terms of saved employees' working hours.

The rest of this paper is structured as follows. In Section II, previous solutions provided in the literature are described. Sections III are devoted to the description of the proposed architecture. In Section IV, we give more details about the data collection, pre-processing, and behavior analysis module. In Section V, a real-life case study is presented exploring the relevance of the proposed approach, in addition to the ML model performance evaluation and experimental results showing the improvement in transaction processing. In Section VII, a conclusion is presented, followed by a discussion with suggestions for future work.

II. RELATED WORKS

Many solutions were proposed in the literature to manage and optimize banking business processes and applications. Some of them implemented the traditional role-based access control (RBAC), while others used local access control files, discretionary and mandatory access control models (DAC and MAC), and context concept on top of the traditional RBAC. Moreover, many approaches have addressed optimizing the business processes using machine learning models. In this regard, we provide in the following a brief overview of the literature addressing the aforementioned areas while highlighting their limitations and distinguishing our proposed work.

In a mandatory access control matrix (MAC), a super-user or supervisor defines the access level of each subject [15]. The sensitivity of objects at each level is defined by the procedure or object owner. In [16], the authors stated that in discretionary access control (DAC), the owner establishes the relation between subjects and objects. DAC is implemented in the access control matrix (ACM) and Take-Grant systems. MAC is implemented in Bell-LaPadula (BPL) and Chinese-Wall systems. Lack of flexibility is the main drawback when implementing MAC or DAC [3]. The popular access control model, role-based access control (RBAC) [17], aimed to assign permissions to users through roles as an intermediate layer to access resources. Users are assigned to roles, and permissions are assigned to roles. The user role relation and role permission relation constitute the user permission relation. The later facilitates the access control management. RBAC facilitates the role of engineering tasks and access control management by grouping users into roles. Maintenance takes place on the role level instead of maintaining and managing permissions on the user or object level as direct relation similar to access control matrix (ACM), access control list (ACL), or user capability methodologies. RBAC works well in large enterprises that have a large number of users. Banks and other environments widely used the local access control files model before 1990. It defines the users' and employees' access rights and is placed on each application host. After 1990, banks started moving to RBAC models since the local access control model is time-consuming, difficult to use and maintain, and error-prone. As discussed by Schaad *et al.* [2], local access control files are challenging to maintain compared to role-based access control. RBAC in banks and other environments is used to move from direct user permission relation to RBAC model based on either bottom-up or/and top-down role engineering. The top-down approach configures RBAC based on business processes and security policies. The bottom-up approach finds RBAC configuration suitable with respect to the existing user permission assignments [17]. However, traditional RBAC must be updated manually and does not take user history, experience, behavior, geographic locations and time into consideration.

Aiming to suit a new environment and provide more flexibility, the context concept was introduced to RBAC by many researchers. Covington *et al.* [4] introduced in his Generalized Role-Based Access Control model a context-oriented environment role in addition to the traditional subject and object roles. Bertino *et al.* [6] proposed the temporal RBAC that considers the time dimension. Uzun *et al.* [18] proposed a temporal RBAC that defines the user to role, permission to role and role to role assignments as temporal relations. GRBAC is based on temporal RBAC that considers the time dimension. Joshi *et al.* [19] proposed a GRBAC method that can express a wider range of temporal constraints like periodic and duration constraints on roles, user roles, and role permission assignment. Perlasca *et al.* [7] proposed the GEO-EBAC model that introduces the spatial extension to traditional RBAC. Park *et al.* [5] benefited from the context information and added a context role to the RBAC model mapped with an organizational role to suggest a relational based access control model. Aich *et al.* [20] proposed the spatiotemporal RBAC (STARBAC) that takes into consideration time and location constraints in addition to the traditional RBAC. Zedan and Al-Sultan proposed in [21] a policy-based approach that defines policies as compositional rules where policies are analyzed and verified at run time. Md Emanual Kabir and Hua Wang [22] proposed a conditional based access control model that enables access to data based on the conditional intended purpose. This model focused more on customer information privacy than security and access. Wang *et al.* proposed in [23] a new offline electronic cash scheme that provides flexible anonymity and untraceable ability using the RBAC management technique. The same authors proposed in [24] privacy-preserving approach for task recommendation systems. The proposed method ensures the duty separation concept. Wand *et al.* provided in [25] a role-based group delegation framework (RBGDF) to address the sharing of content in virtual universities environment. A user is assigned to one or multiple sessions. The session maps the user to one or multiple roles. Therefore, the user can be either an original user of a role or a delegated one of another role. All these approaches offered important context improvement compared to the traditional access techniques. However, they still do not take user history, behavior, and performance into consideration and do not embed intelligent analysis for updating policy and procedural control.

In another context, several approaches in the literature have been proposed to optimize business processes using machine learning. In the sequel, we summarize the ones related

to our approach. Folino *et al.* and Francescomarino *et al.* proposed in [26], [27] predictive business process monitoring approaches. The non-process-aware schemes are based on machine learning techniques using clustering methods to classify the business process during run time while considering similarities with historical logs, data properties, sequence of events, and frequency of occurrences. Venerich *et al.* proposed in [28] a predictive business monitoring method at run time using SVM based on the process remaining time, risk probability, and next event. Marquez *et al.* provided in [11] a Decision Tree-based approach to predict business process status during run time based on a sequence of events and event data information. Conforti *et al.* offered in [12] a predictive monitoring approach based on decision tree technique while taking into consideration the probability of risk. The risk is determined based on resources, activities sequence, and future events. Leontjeva presented in [29] a similar approach that analyzes input information like sequence and order of events in addition to the frequency of appearance. This method is based on the Hidden Markov Model and Random Forest techniques. The closest to our proposal is the behavior-based access control method proposed by Mohammad *et al.* in [8]. The provided method analyzes the user history in terms of time of use, ordering of performed actions on the system by the user, and association constraints. Then, it compares this behavior to a predefined set of rules and adjusts them accordingly. However, to the best of our knowledge, none of the aforementioned literature work has yet addressed the adaptation and customization of roles and systematic reviews and updates of the control policies and procedures in banking sector based on employee/user historical usage and performance analysis using machine learning techniques. Our approach considers the context and behavior factors related to users' experience and performance in order to improve and customize the deployed role-based access and policy control for optimizing the teller banking business process.

III. APPROACH OVERVIEW AND MODEL DESCRIPTION

In this section, we describe the approach overview and the interactions between its components. The architecture is depicted in Fig.1, which represents the old business process for tellers compared to our proposed solution. In the old process, tellers receive transaction requests from customers to be processed based on their roles and limits. If tellers are not prompted access to process a given transaction, they ask for the supervisor's intervention to complete the processing. A predefined control policy component is usually used to store static policies that define tellers' roles and limits. In case the performance of the teller improves, the traditional in-place assessment has to be performed and an update for the static policies has to take place manually to change the role and grant him/her more privileges. Meanwhile, tellers will continue sending requests for approval to the supervisor. Similarly, in case the teller is granted the privilege to execute a transaction that he is no more eligible to do, the static control

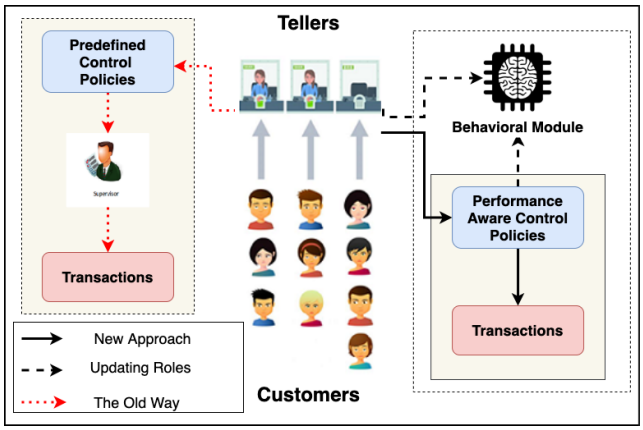


FIGURE 1. The Difference Between Old Performance Evaluation Process (Left) v.s. The New Proposed Process (Right).

policies allow him until it is updated. The drawback of this approach is the need to manually assess the teller performance with no future prediction and update the predefined control policies component to reflect the new roles of tellers, which in turn costs time and human efforts.

In order to automate both the performance assessment with future behavior prediction and the dynamic updates to the policies, we propose the architecture illustrated on the right side of Fig.1. The performance-aware control policy dynamically updates the control policies to grant tellers access to functionalities they want based on their predicted work behavior. Following our approach, tellers get the authorization for executing a particular transaction from the performance-aware control policies. This component updates its policies dynamically by invoking the prediction of the intelligent behavior analysis module, which predicts a suitable update on the teller role by studying its history of performance. Following this approach, tellers' roles are dynamically upgraded and downgraded without the need to reconfigure the whole policy manually. In the sequel, we describe two of the main components in the proposed architecture, which are the Behavior Analysis Module and Performance-Aware Control Policies.

A. BEHAVIOR ANALYSIS MODULE

The history of tellers' performance is collected and stored in data storage. This data contains info about the transaction details, errors, transaction complexity, and authorization. The behavior analysis module embeds a machine learning classifier used for evaluating users' roles while considering their work behaviors and performance history and classifies them into three categories: (1) Downgraded Limit, (2) As-Is Limit and (3) Upgraded Limit. More details about the data collection, preparation, and behavior analysis module are provided in Section IV.

B. PERFORMANCE-AWARE CONTROL POLICIES

The performance-aware control policies module generates dynamically and systematically the updated policies

reflecting the result of the intelligent work behavior and performance classifier for embedding them in the tellers business process. The policy is updated based on predefined downgrade and upgrade margins while taking into consideration the old teller limit and the behavior module prediction. For instance, the new teller limit will be getting the old limit and multiplying it by the upgrade margin in case the model predicted an upgrade and vice versa. For achieving the systematic update of policies in the banking teller business process to reflect the new roles of tellers, we propose the use of a process independent language such as the eXtensible Access Control Markup Language (XACML) [30].

XACML provides an authorization mechanism that can be used to express and evaluate the updated policies. The choice of XACML is due to its capability of decoupling policies from enforcement, which facilitates the policy adaptation required in our model. Our authorization component embeds five different sub-components. The first one is the policy enforcement point (PEP) that receives a request to access functionality for executing a transaction. PEP then forwards this request to the policy decision point (PDP), which is responsible for assessing the incoming request and make a decision to either authorize operation or not. This decision is sent back to the PEP. The definition of the dynamic policies based on the authorization language is stored in the policy retrieval point (PRP), which is called by the PDP to evaluate the policies. Because the policy evaluation relies on the behavior analysis module, the policy information point (PIP) is responsible of invoking it to get the predictions of the new limits for the teller. As alternative to XACML, several extensions such SBA-XACML and its related analysis were proposed to make the policy evaluation faster using sets based languages in [31]–[33]. Moreover, it is important to mention that the use of aspect-oriented programming may improve the model by allowing dynamic integration of the policies changes within XACML and business processes [34], [35].

IV. BEHAVIOR ANALYSIS MODULE STUDYING TELLERS PERFORMANCE

In this section, we discuss the process of building the machine learning behavior analysis module described in the proposed architecture of Section III. It constitutes the core behavior assessement and decision module for the performance-aware control policies to dynamically update its authorization rules and allow tellers processing transactions based on their predicted role. In order to study the teller's behavior, any data from a banking environment describing his/her performance from supervisors' logs is helpful. The collected data is then subject to pre-processing and consequently modeling for classifying tellers' new roles and limits based on their history of behaviors and performance. In the sequel, we detail the procedure of data pre-processing and describe the adopted ML models and intended predictions.

A. DATA COLLECTION AND PRE-PROCESSING

Transactions details are extracted from the core banking system database like the user, role, date, and transaction type. Errors are extracted from the system journals and logs. The transactions performed by tellers are classified under two main categories: transactions with low complexity and transactions with high complexity. Transactions with low complexity are usually easy to process with few numbers of input fields. An example of low complexity transactions can be transfer between accounts with the same nature, credit transactions, cash withdrawal, etc. On the other hand, examples of transactions with high complexity can be draft cheque cash and account, issue draft cash and account, exchange cash, etc. The bank management specifies the Margin Rate and Limit Value. The Margin Rate is the rate by which the limit may be increased or decreased. The Limit Value is the transaction limit defined per role. Transactions above the given limit require an authorization.

Thereafter, the first step of data pre-processing results in the below listed features. We put the actual feature name in double quotes.: (1) user role or ''Role Id'', (2) total number of transaction executed by all tellers belonging to the same role or ''Total Role Trx'', (3) number of transactions executed by a teller with high complexity or ''Total Trx Comp High'', (4) number of transaction executed by teller with low complexity or ''Total Trx Comp Low'', (5) total number of errors made by all tellers in the role or ''Total Err'' (6) number of low complexity errors made by a teller or ''Total Err Comp Low'' and (7) number of high complexity errors made by a teller or ''Total Err Comp High''. Finally, the label of the data either to ''downgrade'' the teller role, ''keep as is'', or ''upgrade''. These are the seven main features to be extracted from datasets serving the same purpose. Feature selection can be applied based on the correlation of the data at hand.

B. MULTI-CLASS MACHINE LEARNING CLASSIFIER

Given the behavior and performance of the tellers from the data described in the previous section, a multi-class classifier ML model is built to decide on a newly assigned limit for the teller. We rely on Scikit-Learn python library [36] to build the SVM multi-class classifier, in which we solve an optimization problem to find a hyperplane having the maximum margin to the support vector [37]. We then use the decision rule to predict a class for the incoming samples. When the data is non-linearly separable, we use kernels to project the data into a high dimensional space to find a hyperplane that better separates the data. In the context of our problem, we use linearSVC [36] with RBF kernel to get the highest accuracy compared to other classifiers. More details about the performance measure of SVM compared to other models are provided in the experimental results of Section V. The output of the model is to predict one of the classes to downgrade (1), keep as is (0), or upgrade (2). As described in Section III, the PIP of the performance-aware control policies

TABLE 1. Correlation Results of Features.

Feature	Role ID	Total Role Trx	Total Role Err	Total Trx Comp Low	Total Trx Comp High	Total Err Comp Low	Total Err Comp High
Role ID	1	0.66	0.66	0.37	-0.12	0.27	0.28
Total Role Trx	0.66	1	0.99	0.57	0.12	0.52	0.52
Total Role Err	0.66	0.99	1	0.57	0.12	0.52	0.53
Total Trx Comp Low	0.37	0.57	0.57	1	0.38	0.2	0.78
Total Trx Comp High	-0.12	0.11	0.11	0.38	1	0.53	0.49
Total Err Comp Low	0.27	0.52	0.52	0.9	0.53	1	0.81
Total Err Comp High	0.28	0.52	0.53	0.7	0.49	0.8	1

TABLE 2. Excerpt of the Dataset.

Role Id	Total Role Trx	Total Trx Comp Low	Total Trx Comp High	Total Err Comp High	Label
1	400	47	65	1	0
4	688	91	216	1	1
1	420	133	120	1	2
7	13732	385	91	1	0
5	1354	5	1	0	1

launches the SVM model to get predictions of the tellers' new limit for making the authorization decisions of the coming transactions.

Other ML classifiers, such as the Random Forest and multinomial logisitic regression classifiers, are also valid for our problem. Random Forest is an ensemble supervised machine learning technique based on decision trees as the basic building block of Random Forest algorithm. Multiple decision trees are generated by random forests with randomization [38]. The decision tree contains a root, internal, and terminal (leaves) nodes. Each internal node is the root of the sub-tree. Internal nodes correspond to features, and branches represent a test attribute binary partition. In random forest, splitting of attributes at the level internal nodes is performed at the attribute that has the highest Gini Index [38]. On the other hand, Multinomial Logistic Regression is built on top of logistic regression binary classifier in order to enable multi-class classification. In this model, we try to minimize the cost function by applying gradient descent to reach a global minimum resulting in the intended model [39]. A comparison of performance results for these different ML models is provided in Section V-B. After getting the classification decision by the PIP for the teller, the PEP updates the new limit of the teller by applying predefined margins to downgrade or upgrade the teller role as described in Section III.

V. APPROACH VALIDATION: EXPERIMENTS AND PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed scheme using real-life data collected from a banking environment. First, we describe the collected data and perform pre-processing and feature selection. Second, we describe the configuration of different machine learning models and compare their performance measures. Finally, we present a real-life case study for one bank branch and demonstrate the improvement achieved in the authorization time of tellers' transactions by adopting our approach.

A. DATASET FROM REAL BANKING ENVIRONMENT

There are no available approaches, datasets, or real-life tests in the literature within the same scope of our proposed work in order to validate and compare our results. Accordingly, we discuss in this section a real-life data collected by us from working environments in several banks (names hidden for privacy and confidentiality purposes). The collected dataset has 303,710 transactions distributed over 13 types of transactions and performed by 47 tellers under seven roles and extracted over biweekly periods during 12 months of the year 2019. The transactions with low complexity constitute on average around 83% of the workload for all tellers, while the transactions with high complexity constitute 17% on average.

After performing the data pre-processing task described in Section IV, we get the data with seven features shown in Table 1. In order to decide on the proper feature selection based on the collected data, the Pearson correlation scores are calculated to generate values between -1 and 1 for describing the linear relation between the data features. The correlation values of different combinations of data are illustrated in Table 1. We consider the features having values greater than 0.8 as strongly correlated, therefore one of them can be dropped. Based on the correlation results of the collected data, the *Total Role Err* is highly correlated with *Total Role Trx*. Similarly, the *Total Err Comp Low* is highly correlated with *Total Trx Comp Low*. Therefore, the *Total Role Err* and *Total Err Comp Low* are eliminated to result in the five features shown in Table 2. We show an excerpt of the resulting dataset describing the historical performance of some tellers during a random bi-week period in Table 2. Furthermore, we show the distribution of the number of transactions and their error rate for tellers 1 and 10 during the whole year, as shown in Figures 2 and 3 respectively. As seen in these figures, we validate that the data used is generated from a real banking environment, where the error rate (grey line) is low compared to the number of transactions processed bi-weekly. It is also important to note that the whole dataset is balanced with the following percentages for the three labels: 44.65% for label 0 (keep as is), 34.1% for label 1 (downgrade), and

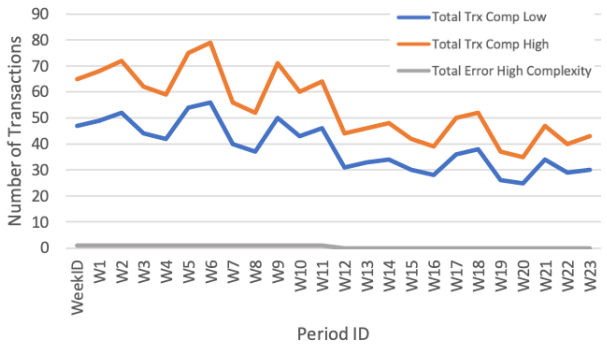


FIGURE 2. Transactions and Errors Performed by Teller 1.

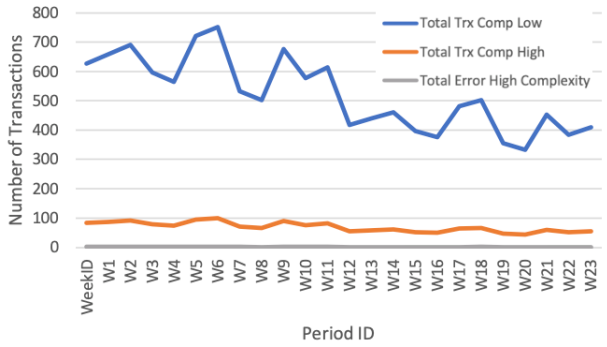


FIGURE 3. Transactions and Errors Performed by Teller 10.

21.28% for label 2 (upgrade). In our dataset, labels were assigned manually by a banking expert in charge of evaluating the performance of each teller.

B. ML MODEL EVALUATION

Our behavior analysis model is based on a multi-class classification ML model. After the data preparation phase, we experimented with different ML classifiers in order to adopt the one with the best performance. In this section, we describe the configuration and compare the performance of the following models: Multinomial Logistic Regression (MLR) [39], Random Forest (RF) [40], and Support Vector Machine (SVM). Using Scikit-learn library [36], we were able to implement these different models. MLR was configured to use L1 as the norm in penalization, and 100 as the inverse regularization strength. The RF model was configured with 100 trees. Finally, we used the RBF kernel to build the linear SVM with a regularization parameter of 1000, and gamma value of 0.1. We also used grid search functionality provided by Keras to get the best parameters listed above. The models were trained and tested using Random permutation cross-validation KFOLD method. For cross validation, we divided the data into 60% training, 20% validation and 20% testing, with a random generator of 100 and 14 re-shuffling and splitting iterations. We considered the F1-score, accuracy, precision, and recall to compare the performance of the different models. As shown in Table 3, SVM achieved

TABLE 3. Comparison of performance measures for the different ML classifiers.

Mode	F1 score	Accuracy	Precision	Recall
MLR	56.03	56.03	56.03	56.03
RF	85.75	85.75	85.75	85.75
SVM	88.07	88.07	88.07	88.07

an accuracy of 88.07% and outperformed both RF with an accuracy of 85.75% and MLR with an accuracy of 56.03%.

C. AUTHORIZATION TIME EVALUATION

As presented in the list of contributions, our approach allows reducing the time needed to perform transactions at branches and consequently provide faster service to customers. This is done by reducing the number of authorizations for transactions made by the upgraded roles without affecting the assigned constraints satisfied by the system configuration. In this context, we used part of the data related to seven tellers and belonging to only one branch. We then performed an analysis to show the time consumed and gained by applying the proposed scheme.

For the extracted data, the total number of performed transactions is 47,618. The transaction limit set in the studied sample is $2,000, and the number of thrown authorizations is 18,973. Our real-life experiment performed on-site at the bank and based on logs and inputs from banks' officers shows that a transaction with a low complexity level like cash deposit takes on average 90 seconds, while a transaction with high complexity level like clean payment takes on average 360 seconds, including the time needed for authorization. The authorization takes on average 35 seconds if one authorization is sent at a time, 70 seconds if one authorization request is waiting in the queue, and 105 seconds if two transactions are waiting in the queue.

Table 4 explores the advantages of our proposed approach by illustrating the % of reduction in the number of required authorizations per limit upon increasing the role limit progressively by our model from $2,000 to $30,000. Moreover, it depicts its corresponding time gain in hours per limit per month upon increasing the limit by our model while considering the cases where no authorization is held in the queue, one authorization is held in the queue, and two authorization requests are held in the queue. For instance, if the transaction limit is increased to $10,000, 8,200 authorization requests are thrown, which means 56.78% of authorization requests are reduced. Accordingly, this reduction in the number of authorization requests reduces the time needed to serve customers and process their transactions. Therefore, in one branch, 9.52 hours are saved per month if no authorization request is waiting in the queue, 19.04 hours are saved per month if one authorization request is waiting in the queue, and 28.57 hours are saved per month if two authorization requests are waiting in the queue. It is worth mentioning that the aforementioned gain is only in one branch of a bank. Accordingly, the overall

gain might be significantly higher and may vary according to the workload per branch and bank size.

Furthermore, Figures 4, 5 and 6 illustrate the results of the statistical analysis on the data provided in Table 4. Figure 4 shows the relation between the number of requested authorizations and the transaction limit set. For instance, the number of authorization requests decreases from 18,973 when the limit is \$2,000 to 4,371 when the limit becomes \$30,000. Accordingly, the reduction in authorization requests leads to a reduction in the time needed to process the transactions.

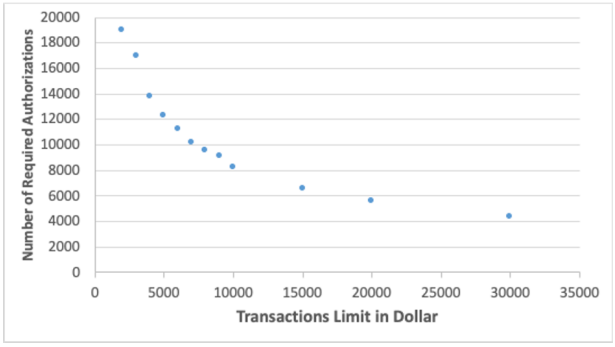


FIGURE 4. Number of Authorization Per Limit.

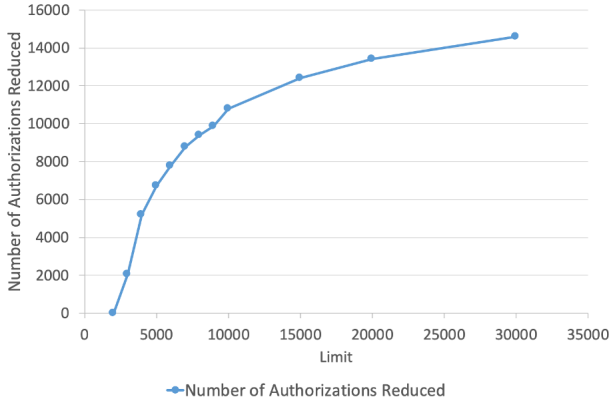


FIGURE 5. Number of Authorizations Reduced per Limit With an Initial Number of Authorization of 18,973.

Figure 5 shows the reduction in the number of required authorization when our model increases the transaction limit. For instance, around 11,000 authorization requests are reduced if the limit is increased from \$2,000 to \$10,000. Accordingly, the time needed to process the transactions is also be reduced.

Finally, Figure 6 shows the average time reduced in hours per month. The curve in blue represents the relation between the transaction limit and time reduced if one authorization request is sent at a time, starting from 5 hours for \$5,000 limit and reaching 11 hours for \$30,000 limit. The orange curve represents the relation between the transaction limit and time reduced if one authorization request is waiting in the queue when an authorization request is being processed, starting from 10 hours for \$4000 limit and reaching 25 hours for \$30,000 limit. The gray curve represents the relation between the transaction limit and time reduced if two authorization requests are waiting in the queue when an authorization request is being processed, starting from 14 hours for \$4,000 limit and reaching 38 hours for \$30,000 limit.

FIGURE 6. Time Reduced in Hours per Month.

VI. DISCUSSION, LIMITATIONS, AND DIRECTIONS

To elaborate more, we offer in the sequel an in-depth discussion on the value of the research work and its potential impacts on industry, managerial procedures, and social aspects. In fact, machine learning techniques proved to be very successful in diverse areas, including banking business processes like credit and loan risk management, online web business processes and services, and e-commerce. However, the literature is still lacking the integration of such intelligence within the daily banking business operations due to its high-cost implications in case of failures. Here it comes the importance and value of our approach, which provides a showcase on embedding intelligent learning of employees' historical performance for optimizing the daily banking processes (i.e., teller in our approach) and reducing the overhead of some operations that can be automated and/or avoided through systematic amendment of control policies. Our methodology is very promising scheme from a managerial perspective and trusted to be implemented in the current banking system since it is embedding a two-sided incentive-based reward/punishment mechanism (e.g., upgrade or downgrade policy) and a possibility to enforce unforgeable constraints not to bypass through the systematic model. In terms of social implications, the applied approach may not increase the risk since critical constraints can still be incorporated, while at the same time, it can reduce the waiting time for both clients and employees. Indeed, the presented case study, based on a real-life banking environment, supports our claims and explores the relevance of the logical results induced by our proposed model based on the performance analysis of seven tellers belonging to one branch. Moreover, the provided experiments performed on 47,618 real-life transactions illustrates the potential of our scheme by reducing the time per transaction and consequently, the total number of employees' hours.

TABLE 4. Reduction in Number of Required Authorizations and Gain in Time.

Transaction Limit in Dollars	Number of Required Authorizations Per Limit	Percentage of Reduced Authorization Per Limit	Time Reduced in Hours Per Limit Per Month (based on 35s Per Authorization)	Time Reduced in Hours Per Limit Per Month If 1 Transaction is Waiting in Authorization Request Queue (based on 70s Per Authorization)	Time Reduced in Hours Per Limit Per Month If 2 Transactions are Waiting in Authorization Request Queue (based on 105s Per Authorization)
2,000	18,973	0.00%	0.00	0.00	0.00
3,000	16,924	10.80%	1.81	3.62	5.43
4,000	13,767	27.44%	4.60	9.20	13.81
5,000	12,262	35.37%	5.93	11.86	17.80
6,000	11,193	41.01%	6.88	13.75	20.63
7,000	10,185	46.32%	7.77	15.54	23.30
8,000	9,594	49.43%	8.30	16.58	24.87
9,000	9,085	52.12%	8.74	17.48	26.22
10,000	8,200	56.78%	9.52	19.04	28.57
15,000	6,548	65.49%	11.00	22.00	33.00
20,000	5,560	70.70%	11.86	23.71	35.57
30,000	4,371	77.00%	12.91	25.81	38.72

The main limitations of the proposed scheme are the inability of the ML model to adapt to the possible dynamic changes in the tellers performance, and the shortage of similar approaches and datasets available for research and comparison. However, the achieved results are promising to build on top and identify several directions for future work. First, different intelligent and machine learning techniques can be investigated to possibly achieve better accuracy and performance for various business processes within banking systems. In addition, Reinforcement Learning may be used to study the behavior of tellers and make online role adjustment decisions [41]. Second, several tracks and techniques can be investigated to realize and deploy the proposed dynamic approach in the current policy-based languages used for business process and service management [42].

VII. CONCLUSION

In this paper, we addressed problems related to the current banking teller systems and processes, in which performance is assessed using on-time traditional methods and policies and roles assigned to employees are accordingly controlled by static models and configurations reviewed manually by managers. This leads to additional authorization overhead that may not always be needed if the performance of the employee has shown enough stability over historical time. In this context, we proposed a new scheme embedding an Intelligent Behavior Analysis and Performance-Aware Adaptation of Roles and Control Policies for optimizing the banking daily-operations process of tellers. Although the teller banking process is used for modeling and experimenting, the proposed scheme may be extended and applied to other banking business processes. The elaborated performance-aware control policies can be converted to authorization mechanism based on the context used. In addition, the behavior analysis model utilizes a machine learning scheme using SVM, which extends to assessing the deployed process and control policies and updating them systematically with new roles and authorization levels based on employees' behaviors and work history while respecting the system constraints. In other words, our approach allows classifying employees based on their performance. It then suggests modifying the authorization levels and limits accordingly, which leads to a reduction in the authorizations request during each transaction and consequently, the total processing time. The presented case study and experiments, performed on a dataset collected from real-life banking environment, illustrate the feasibility and efficiency of the proposed approach in terms of accuracy of the classification model and reduction of the required authorizations and employees' working hours.

REFERENCES

[1] M. Rouse, "Four eyes principal," Techtarget, Newton, MA, USA, Tech. Rep., 2013.

[2] A. Schaad, J. Moffett, and J. Jacob, "The role-based access control system of a European bank: A case study and discussion," in *Proc. 6th ACM Symp. Access Control Models Technol.*, 2001, pp. 3–9.

[3] K. I. Kim, H. J. Ko, H. S. Hwang, and U. M. Kim, "Context RBAC/MAC access control for ubiquitous environment," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Bangkok, Thailand: Springer, 2007, pp. 1075–1085.

[4] M. J. Covington, M. J. Moyer, and M. Ahamad, "Generalized role-based access control for securing future applications," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GIT-CC-00-02, 2000.

[5] S.-H. Park, Y.-J. Han, and T.-M. Chung, "Context-role based access control for context-aware application," in *Proc. Int. Conf. High Perform. Comput. Commun.* Munich, Germany: Springer, 2006, pp. 572–580.

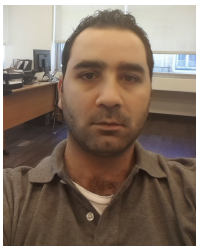
[6] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A temporal role-based access control model," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 191–233, 2001.

[7] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca, "GEO-RBAC: A spatially aware RBAC," in *Proc. 10th ACM Symp. Access Control Models Technol. (SACMAT)*, New York, NY, USA, 2005, pp. 29–37.

[8] M. H. Yarmand, K. Sartipi, and D. G. Down, "Behavior-based access control for distributed healthcare systems," *J. Comput. Secur.*, vol. 21, no. 1, pp. 1–39, Feb. 2013.

[9] M. H. Jansen-Vullers, M. Netjes, and H. A. Reijers, "Business process redesign for effective E-commerce," in *Proc. 6th Int. Conf. Electron. Commerce (ICEC)*, 2004, pp. 382–391.

[10] F. Niedermann, S. Radeschütz, and B. Mitschang, "Business process optimization using formalized optimization patterns," in *Proc. Int. Conf. Bus. Inf. Syst.* Poznań, Poland: Springer, 2011, pp. 123–135.
[11] A. M. Chamorro, M. Resinas, and A. RuizCortes, "Towards a general architecture for predictive monitoring of BPS," *Jornadas Ciencia Ingeniera Servicios*, vol. 220, pp. 29–32, Sep. 2016.
[12] R. Conforti, M. de Leoni, M. La Rosa, W. M. P. van der Aalst, and A. H. M. ter Hofstede, "A recommendation system for predicting risks across multiple business process instances," *Decis. Support Syst.*, vol. 69, pp. 1–19, Jan. 2015.
[13] Y. Guo, D. Tao, W. Liu, and J. Cheng, "Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 617–627, Apr. 2017.
[14] R. Hussein, J. Lin, K. Madden, and Z. J. Wang, "Robust recognition of human activities using smartphone sensor data," in *Proc. Int. Conf. Frontiers Adv. Data Sci. (FADS)*, Oct. 2017, pp. 92–96.
[15] J. M. Kizza, *Guide to Computer Network Security*. Cham, Switzerland: Springer, 2009.
[16] G. Pernul, "Security constraint processing during multilevel secure database design," in *Proc. 8th Annu. Comput. Secur. Appl. Conf.*, 1992, pp. 75–84.
[17] M. Frank, M. J. Buhman, and D. Basin, "Role mining with probabilistic models," *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 4, pp. 15:1–15:28, 2013.
[18] E. Uzun, V. Atluri, J. Vaidya, S. Sural, A. L. Ferrara, G. Parlato, and P. Madhusudan, "Security analysis for temporal role based access control," *J. Comput. Secur.*, vol. 22, no. 6, pp. 961–996, Dec. 2014.
[19] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 1, pp. 4–23, Jan. 2005.
[20] S. Aich, S. Sural, and A. K. Majumdar, "STARBAC: Spatiotemporal role based access control," in *Proc. OTM Confederated Int. Conf. Move Meaningful Internet Syst., CoopIS, DOA, ODBASE, GADA, IS, OTM*. Berlin, Germany: Springer-Verlag, 2007, pp. 1567–1582.
[21] H. Zedan and S. Al-Sultan, "The specification and design of secure context-aware workflows," *Expert Syst. Appl.*, vol. 86, pp. 367–384, Nov. 2017.
[22] M. E. Kabir, H. Wang, and E. Bertino, "A conditional purpose-based access control model with dynamic roles," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1482–1489, Mar. 2011.
[23] H. Wang, J. Cao, and Y. Zhang, "A flexible payment scheme and its role-based access control," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 425–436, Mar. 2005.
[24] J. Shu, X. Jia, K. Yang, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE Trans. Services Comput.*, early access, Jan. 10, 2018, doi:10.1109/TSC.2018.2791601.
[25] H. Wang, Y. Zhang, and J. Cao, "Effective collaboration with information sharing in virtual universities," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 6, pp. 840–853, Jun. 2009.
[26] F. Folino, M. Guarascio, and L. Pontieri, "Discovering context-aware models for predicting business process performances," in *Proc. OTM Confederated Int. Conf. Move Meaningful Internet Syst.* Rome, Italy: Springer, 2012, pp. 287–304.
[27] C. Di Francescomarino, M. Dumas, C. M. G. Federici, F. Maggi, and W. Rizzi, "Predictive bp monitoring framework with hyperparameter optimization," in *Proc. Adv. Inf. Sys. Eng. (CAISE)*, 2016, pp. 361–376.
[28] I. Verenich, M. Dumas, M. La Rosa, F. M. Maggi, and C. Di Francescomarino, "Minimizing overprocessing waste in business processes via predictive activity ordering," in *Proc. 28th Int. Conf. Adv. Inf. Syst. Eng.* Ljubljana, Slovenia: Springer, 2016, pp. 186–202.
[29] A. Leontjeva, R. Conforti, C. Di Francescomarino, M. Dumas, and F. M. Maggi, "Complex symbolic sequence encodings for predictive monitoring of business processes," in *Proc. Int. Conf. Bus. Process Manage.* Innsbruck, Austria: Springer, 2016, pp. 297–313.
[30] *Extensible Access Control Markup Language (XACML) Version 3.0*, Standard xacmlv3.0, OASIS, 2005. [Online]. Available: https://www.oasis-open.org/standards#xacmlv3.0
[31] H. Jebbaoui, A. Mourad, H. Otrok, and R. Haraty, "Semantics-based approach for detecting flaws, conflicts and redundancies in XACML policies," *Comput. Electr. Eng.*, vol. 44, pp. 91–103, May 2015.
[32] A. Mourad and H. Jebbaoui, "SBA-XACML: Set-based approach providing efficient policy decision process for accessing Web services," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 165–178, Jan. 2015.
[33] A. Mourad, H. Tout, C. Talhi, H. Otrok, and H. Yahyaoui, "From model-driven specification to design-level set-based analysis of XACML policies," *Comput. Electr. Eng.*, vol. 52, pp. 65–79, May 2016.
[34] S. Ayoubi, A. Mourad, H. Otrok, and A. Shahin, "New XACML-AspectBPEL approach for composite Web services security," *Int. J. Web Grid Services*, vol. 9, no. 2, pp. 127–145, 2013.
[35] A. Mourad, M.-A. Laverdière, and M. Debbabi, "A high-level Aspect-oriented-based framework for software security hardening," *Inf. Secur. J., Global Perspective*, vol. 17, no. 2, pp. 56–74, May 2008.
[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[37] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.
[38] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[39] J. Starkweather and A. K. Moske. (Sep. 10, 2011). *Multinomial Logistic Regression*. [Online]. Available: http://www.unt.edu/rss/class/Jon/Benchmarks/MLR_JDS_Aug2011.pdf
[40] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
[41] P. Farhat, H. Sami, and A. Mourad, "Reinforcement R-learning model for time scheduling of on-demand fog placement," *J. Supercomput.*, vol. 76, no. 1, pp. 388–410, 2020.
[42] H. Tout, A. Mourad, C. Talhi, and H. Otrok, "AOMD approach for context-adaptable and conflict-free Web services composition," *Comput. Electr. Eng.*, vol. 44, pp. 200–217, May 2015.
[43] B. Tay, *Machine Learning Based Approaches for Intelligent Adaptation and Prediction in Banking Business Processes*. Beirut, Lebanon: Lebanese American Univ., 2018.



BILAL TAY received the B.S. degree in management information system and the M.Sc. degree in computer science from Lebanese American University, Beirut, in 2009 and 2018, respectively.

AZZAM MOURAD (Senior Member, IEEE) is currently an Associate Professor of computer science with Lebanese American University and also an Affiliate Associate Professor with the Software Engineering and IT Department, Ecole de Technologie Superieure (ETS), Montreal, Canada. He has served/serves as an Associate Editor of the IEEE TRANSACTION ON NETWORK AND SERVICE MANAGEMENT, the IEEE NETWORK, the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, *IET Quantum Communication*, and the IEEE COMMUNICATIONS LETTER. He has also served/serves as an General Chair of IWCMC2020, a General Co-Chair of WiMob2016, and a Track Chair, a TPC member, and a reviewer of several prestigious journals and conferences.