IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 5, 2021, accepted January 9, 2021, date of publication January 18, 2021, date of current version February 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052313

Investigation of Applying Machine Learning for Watch-List Filtering in Anti-Money Laundering

MOHANNAD ALKHALILI, MAHMOUD H. QUTQUT, (Member, IEEE), AND FADI ALMASALHA, (Member, IEEE)
Faculty of Information Technology, Applied Science Private University, Amman 11931, Jordan
Corresponding author: Mahmoud H. Qutqut (qutqut@asu.edu.jo)

This work was supported by the Applied Science Private University in Jordan to cover part of the publication fee.

ABSTRACT Financial institutions must meet international regulations to ensure not to provide services to criminals and terrorists. They also need to continuously monitor financial transactions to detect suspicious activities. Businesses have many operations that monitor and validate their customer's information against sources that either confirm their identities or disprove. Failing to detect unclean transaction(s) will result in harmful consequences on the financial institution responsible for that such as warnings or fines depending on the transaction severity level. The financial institutions use Anti-money laundering (AML) software sanctions screening and Watch-list filtering to monitor every transaction within the financial network to verify that none of the transactions can be used to do business with forbidden people. Lately, the financial industry and academia have agreed that machine learning (ML) may have a significant impact on monitoring money transaction tools to fight money laundering. Several research work and implementations have been done on Know Your Customer (KYC) systems, but there is no work on the watch-list filtering systems because of the compliance risk. Thus, we propose an innovative model to automate the process of checking blocked transactions in the watch-list filtering systems. To the best of our knowledge, this paper is the first research work on automating the watch-list filtering systems. We develop a Machine Learning - Component (ML-Component) that will be integrated with the current watch-list filtering systems. Our proposed ML-Component consists of three phases; monitoring, advising, and take action. Our model will handle a known critical issue, which is the false-positives (i.e., transactions that are blocked by a false alarm). Also, it will minimize the compliance officers' effort, and provide faster processing time. We performed several experiments using different ML algorithms (SVM, DT, and NB) and found that the SVM outperforms other algorithms. Because our dataset is nonlinear, we used the polynomial kernel and achieved higher accuracy for predicting the transactions decision, and the correlation matrix to show the relationship between the numeric features.

INDEX TERMS Anti-money laundering, financial transactions monitoring, machine learning (ML), sanctions screening, watch-list filtering.

I. INTRODUCTION

The Financial Action Task Force (FATF) is an intergovernmental organization that promotes and develops policies to guard the global financial system against money laundering, terrorist financing, and the financing of proliferation of weapons of mass destruction. The FATF recommendations are recognized as the global anti-money laundering (AML) and counter-terrorist financing (CFT) standard [1]. All financial institutions should apply these international regulations to deny services to criminals and terrorists or to distinguish suspicious activities and report them to the authorities to prevent money laundering and terrorist financing [1]. These regulations focus on the source of money and with whom it can be exchanged as it may fall under national security.

Dealing with illegal companies and people will lead to direct fines, and suspend business or harm the institutions reputation. Considering a substantial number of transactions, and the significant amount of illegal entities, it is a must to implement an automated system to assure that financial institutions meet the compliance regulations [2]. Currently,

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

financial institutions perform vast volumes of transactions per day, and there are high chances of missing a suspicious transaction or even transactions. Moreover, while large and medium-sized institutions can hire armies of experienced compliance officers, small-sized firms cannot afford to do the same. With such a limited number of officers and the numerous number of transactions, they need in both cases to work smartly to detect suspicious transactions with minimal false positive or false negative.

Anti-money laundering (AML) regulations require that all financial institutions monitor, investigate, and report any suspicious transaction. It needs to ensure that the sender and beneficiary are not listed in the blacklists as it is prohibited to do business with them. This is an essential duty to avoid money laundering; as banksŕeputation is at stake and will have to pay significant penalties. Hence, a proactive and intelligent system is needed to protect banks and to monitor money transactions. One famous example of the AML penalties is the HSBC bank penalty that occurred in 2012 [3]. The bank paid $1.9 billion because of the weak and insufficient money laundering controls that made the bank to be used to launder a river of drug money flowing out of Mexico. Another example was in 2009, Switzerlandś Credit Suisse Group was fined $536 million because helping Iran and other countries to move billions of dollars through the US banking system [4].

Current software and technologies that are used to monitor money transactions and validate sender and receiver information against the blacklisted names are not smart enough to release or block transactions based on historical data. This leads to a delay in transaction processing time and places the financial institutions at risk [5]. Chartis Research (a leading firm in doing research and analysis on the financial sector and global market for risk technology) asked a question on a survey ''What do you consider to be a technology priority for trader surveillance?'' Financial institutions answered the survey question, and it is highlighting the importance of Artificial Intelligence (AI) and Machine Learning (ML) on banking software. 48% of the survey responses were in the direction of using AI [5].

To this end, we propose a novel automated model for monitoring money transactions by applying ML on the watch-list filtering process and sanctions screening. To the best of our knowledge, this paper is the first research work on automating the watch-list filtering systems. Our proposed model aims to achieve better protection and faster processing time than human-based decisions to minimize the false-positive blocked transactions and human efforts by replacing the traditional rule-based system. The proposed model can be integrated seamlessly with the current watch-list filtering system by direct connection to the Database (DB) used by the software. The Watch-list filtering system DB is updated by the money transactions traffic decision and all related information. This information can be used by the ML-Component to analyze the historical transactions before recommending the final decision.

FIGURE 1. Money laundering stages.

The rest of the paper is organized as follows. We provide background topics related to our paper in Section II. In Section III, the literature related to our research is overviewed. We present our proposed model and its phases in Section IV. The performance evaluation and the test scenarios with their results are carried out in Section V, and followed by the conclusion and future work in Section VI.

II. BACKGROUND

We provide background topics and related concepts to our work in this section.

A. MONEY LAUNDERING

Money laundering is the process of converting illicit money to clean money using different methods to hide the money source [1], as described in Figure 1, there are three main stages of money laundering as follows.

- Placement: Inserting the money into the financial system occurs in this stage; this can be done in different ways like smurfing.
- Layering: In this stage, the money will be transfer between different bank accounts to convert it into another form in further steps. This will help to create a complicated layer that will hide the source of money and make it difficult to track it.
- Integration: Is the process of converting the money into different forms such as vehicles and buildings. After this stage, it will be impossible to track it back to the dirty source.

B. ANTI-MONEY LAUNDERING (AML)

Anti-money Laundering (AML) is a set of actions, laws, procedures, and regulations designed to detect and prevent all practices that lead to generating income through illegal activities [1]. The dirty money should be detected in the first two steps placement and layering. Otherwise, it will be difficult to investigate its source in the third stage. The need for automated software to help in catching suspicious transactions is mandatory. Every financial institution is responsible for having an implementation to apply global regulations. Nowadays, different software systems in the market help banks to detect suspicious transactions to decide which customer is safe to open a business channel with.

The AML solutions have essential elements (illustrated in Figure 2) that work together to fight money laundering and ensure that the financial institutions are not making business

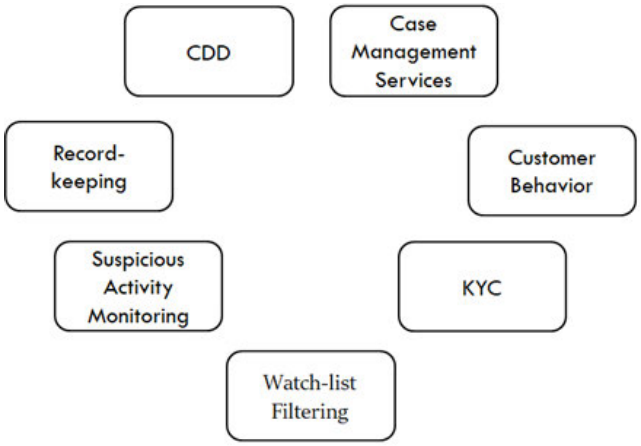


FIGURE 2. Key elements of the AML solutions.

with risky clients [6]. The first element is Know Your Customer (KYC) which is the process of verifying the identity of the clients and assessing the potential risks of making a business relationship with them. The second element is Record-Keeping; this can be achieved by archive historical records for a specific period depends on the regulation of each region. The third is the Suspicious Activity Monitoring element which focuses on monitor client account. The fourth element is Customer Behavior that requires studying the client transactions pattern. The fifth is the Customer Due Diligence (CDD) element that validates the identity of the client before opening an account by checking documents such as passports, photocard driving licenses, and at least two recent utility bills. The sixth and last element is the Watch-list filtering system that we are focusing on in this paper. The Watch-list Filtering system helps financial institutions to apply the required regulations and stay updated with any change or update on the regulations and blacklists' content. This system can filter all transactions and customers against all types of blacklisted entities [6].

C. SWIFT MESSAGES TECHNOLOGY

Society for Worldwide Interbank Financial Telecommunication (SWIFT) is an organization that handles secure transactions and communications between financial institutions worldwide [7]. Each financial institution that uses SWIFT has its code that identifies where the communication originates and what business it is intended for. SWIFT messages are formatted messages that can be used for different purposes. Mt103 is an example of SWIFT messages that is used for single customer credit transfer. The Mt103 message has a standard format to save the needed information to transfer the money from the source to the destination. Essential information includes transaction-ID, sender, and receiver [7].

The SWIFT messages grouped into ten categories as follows. Category 1 is for Customer Payments and Cheques; Category 2 is for Financial Institution Transfers. Category 3 is for Treasury Markets, Foreign Exchange, Money Markets and Derivatives, and Category 4 is for Collections and Cash Letters. Category 5 is for Securities Markets. Category 6 is for Treasury Markets – Commodities, and Category 7 is for Documentary Credits and GuaranteesStandby Letters of Credit. Category 8 is for Travellers Cheques while Category 9 is for Cash Management and Customer Status, and category n is for Common Group Messages.

In this paper, we will be using the Mt103 message from group 1.

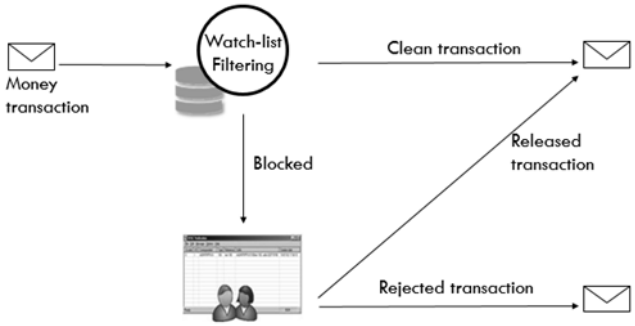


FIGURE 3. Watch-list filtering (current implementation).

D. BLACK LISTS

Blacklists are lists that contain blacklisted people, countries, or other types of entities. There are different types of blacklists, public blacklists that can be used by watch-list filtering system and are published by governments or economic, political, and law enforcement bodies. Also, private blacklists can be created by financial institutions [6]. Every list can have its structure, but usually, they have some vital information such as:

- Name: is the personś full name.
- Name type: can be the main name or Also Known As (AKA).
- Category: the entity type (Individual, Country, Group, etc.).

E. AML SOFTWARE FOR SANCTION SCREENING AND WATCH-LIST FILTERING

Financial institutions implement a watch-list filtering system on their operations system to assist them in monitoring financial transactions and capture any potential risk. The software can be integrated with a bank system to track financial transactions and scan them against a pre-loaded blacklist. Figure 3 shows the current implementations for validating transactions between financial institutions.

The AML software for watch-list filtering is implemented bidirectionally; at sender and receiver institutions. Every party is responsible for validating the transaction information. The AML software implements a string matching algorithm that verifies the transaction information such as names, aka, addresses, and countries against the preloaded blacklists. If the transaction does not generate any alerts, this transaction is considered clean and released by the financial institute.

TABLE 1. Examples of rank calculations.

Scanned Text (Transaction Info)	Matched Text (Blacklist Entity)	Matched Rank
Maik William Jeorge	Maik William Jeorge	100%
Maik William Jeorge	Malek William Jeorge	79%
Maik William Jeorge	Marry Waleed Jeorge	55%

Otherwise, a warning will be created and logged appropriately until a compliance officer investigates it to make a decision; either make it pass or reject it.

F. MATCHING RANK

As part of the AML program implemented in financial institutions, transactions are validated against blacklists using a string-matching algorithm. String matching is based on an exact match or a partial match (not 100% match) between the scanned and blacklist entities. The more accurate the algorithm is, the more the system is effective [6]. For the partial match, it will consider differences between names to calculate the rank value. The current implementation explicitly decides either to release or to block the transaction based on the string matching algorithm results. If the matching rank is low enough, the transaction will be released. Otherwise, a ticket will be reported to the compliance officer for further investigation. As part of the AML program, the string matching algorithm results will be compared with a threshold value defined by the financial institution. If the matching rank is less than the threshold value, the transaction will not be blocked. However, if the matching rank is the same or higher than the threshold value, a detection ticket will be reported [6].

When the threshold value is minimized, more hits will be generated, and more transactions will be blocked; this will increase the false-positive. False-positive detection is detection that should not be blocked. Also, this will increase the required effort to investigate more blocked financial transactions as the program will stop transactions that should not be stopped and declare a faulty alarm. Setting the predefined value to a proper value is critical to avoid skipping compliance transactions (false-negative) or getting a massive number of blocked transactions. This is will either put the financial institution in a risky situation to bypass a risky transaction or causing a delay in money transfer because of the vast number of transactions that are waiting to be investigated. In both cases, the bank will be subject to penalties.

So, the success of the watch-list filtering system depends on the matching algorithm which measures the matching rank while taking into consideration the various possibilities such as spelling variations, phonetic variations, double names, double first names, and alternative first names [8]. Figure 4 explains how the watch-list filtering system utilizes the matching string algorithm to calculate the matching rank. We show an example of how the rank values will be calculated considering the string similarity in Table 1.

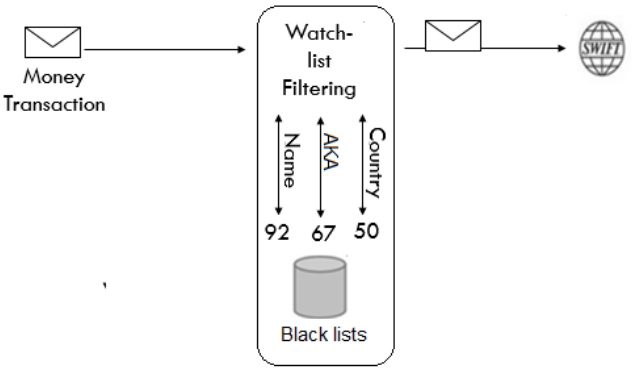


FIGURE 4. Rank calculations.

G. MACHINE LEARNING (ML)

Machine Learning (ML) is one of the Artificial Intelligence (AI) applications that aim to train the machine using historical data. This will allow the machine to understand and categorize the previous data by developing efficient and accurate prediction algorithms [9]. The main categories of the ML are described below.

- Supervised learning: In supervised learning, there should be a set of input attributes and an output value. The algorithm will learn the relation between input and output from the historical data. The algorithm can keep learning from new coming data to enhance accuracy.
- Unsupervised learning: A learning approach that has a set of input attributes but with no output value. The target in this approach is to study the input attributes to find a pattern of similarity between the data to group them.
- Reinforcement learning: This learning approach is different from others; it depends on interacting with the environment and receiving a response for each action.

In this paper, we will be testing and comparing the following algorithms because our paper is considered a first step in validating the idea of applying machine learning on watchlist filtering systems. The selected algorithms were chosen based on those similar problems in the industry that already use them. Hence, our proposed work will open doors for more future works and development.

- Support Vector Machine (SVM): is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane that categorizes new examples. In two dimensional space, this hyperplane is a line dividing a plane into two parts wherein each class lay on either side.
- Naïve Bayes (NB): The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is

widely used because it often outperforms more sophisticated classification methods. We used the Gaussian method in the Naïve Bayes model.

- Decision Tree (DT): supervised ML algorithm that can be used in both regression and classification problems. It works for both categorical and continuous input and output variables. Following the splitting technique to set the input attributes and keep moving to reach the desired output to build the tree using the training data. The tree consists of two entities, the decision nodes, and leaves, where the leaves are the decisions or the outcomes. The decision nodes are where the data is split.

III. LITERATURE REVIEW

In this section, we overview the literature of the AML area, and we conclude the summary of different approaches at the end of this section.

The AML is an enormous industry; researchers and companies are working on different solutions for fighting money laundering and terrorist financing. Also, many companies developed part of these solutions, and other companies officer all of them together as an AML suite [6]. Chartis published a new report titled Artificial Intelligence in Financial Services in 2019 [10]. This report analyses the use of AI in financial institutions and discusses the financial institutionsḱey drivers to implement AI. The report also presents several examples from the industry where AI is a significantly needed element.

The studies showed that AI tools are not yet applied to the decision-making process and there are many hindrances of using AI on regulatory compliance areas. This is because any failure in these areas means significant penalties. On the other hand, the report explains the motivations of using AI in financial services to automate several processes in AML systems and capture more complex trends in the dataset to predict how the system will behave in the future [10].

In the past, AML solutions were based on an operational model that uses simple and easy to code rules. This is usually built by consultants and domain experts who implement these rules on the automated decision process. With time, more rules will be needed which will impact the system performance and accuracy [11]. Recently, researchers start to focus on money laundering control and prevention by automating the monitoring and diagnosing of money laundering schemes to report suspicious activities. Such automation can be done by applying intelligent technologies to deal with all possible money laundering operations [12].

As shown earlier, the AML key elements can be categorized into two groups. The first group is a regulatory compliance area, which is the watch-list filtering element. The second group is related to customer profiles and information that contains the remaining elements (KYC, CDD, suspicious activity monitoring, case management services, record-keeping, and customer behavior).

Researchers have investigated and implemented some ML algorithms using the account transaction patterns of clients that fall under the second group. These algorithms are based on six common aspects of AML solutions. These main aspects are generated from the KYC guideline, which is a standard procedure for every financial institution against AML [11]. In the following, we describe the six AML aspects mentioned earlier.

1) AML typologies: The AML typologies aspect is focused on building a model that defines the AML cases that captured from the historical transactions by following the KYC guidelines, which require a thorough understanding of parties related to the transactions.
2) Link Analysis: The link analysis aspect focus on understanding the relations between entities (e.g., bank accounts), and to analyze the nature of the relations.
3) Behavioral modeling: The behavioral modeling aspect can be achieved by understanding the customers' behaviors based on their transaction activities.
4) Risk scoring: The risk scoring aspect is concerned with ranking all customers and transactions based on potential risk.
5) Anomaly detection: The anomaly detection aspect is the ability to differentiate the unusual transactional behavior of each customer.
6) Geographic capability: The geographic capability aspect depends on strong cooperation between financial institutions among countries to be able to identify money laundering activities across different countries.

An example algorithm for the AML typologies is the CLOPE clustering algorithm. Cao and Do in [13] proposed a technique for applying the grouping approach-based CLOPE calculation to recognize three common money laundering cases. The cases are moving money around, distributing money to many beneficiaries in small amounts, and collecting money from different sources. The experimental outcomes demonstrated that CLOPE could distinguish each suspicious records with only six clusters, without specifying which cluster is related to the money laundering category. This is considered a drawback that makes the approach impractical.

An example of link analysis is the system that supports money laundering detection. Dreewski *et al.* [14] proposed a system consist of three components that support money laundering detection. The first component is clustering; it refers to constructing graphs that represent the flow of money and captures only suspicious money transfers between groups of accounts. The second component is mining for frequent patterns in clusters. The third component is data visualization, and concerns about displaying the result and the transactions flow. The source and destination of a specific banking transaction are required for building the clustering graphs, and it can be challenging to determine the exact destination if the money moves through different banks and countries. Chitra and Subashini in [15] suggested using the expectation-maximization (EM) algorithm as the clustering model in catching fraud for the behavioral modeling aspect. The idea was to use the EM cluster in grouping the data into a similar

cluster by building a model using the historical transaction behavior for each bank. The author did not provide any experimental results on the performance of the algorithm on either synthetic or real datasets. This approach assumed that the customer transactions follow a cretin distribution which is not valid in fact.

For the risk scoring aspect, Sudhakar and Reddy in [11] used the DT algorithm and proposed a two-step loan credibility prediction system that made it easy for the financial institutions to make the right decision to approve or reject the loan request of the customers with the application of DT algorithm. The authors have clarified that credit risk management is critical for a successful bank loan process. Building this model will need five main phases, including problem understanding, data understanding, data filtering, system modeling, and system evaluation.

The suspicious activity reporting using dynamic Bayesian networks (SARDBN) proposed under the anomaly detection aspect by Raza and Haider in [16]. They introduced a mix of clustering and Bayesian networks to identify anomalies in transactions. This approach contains three phases. Firstly, the process of clustering to group customers based on average monthly credit and debit transaction amount, average monthly credit and debit transaction frequency, and the time between the consecutive transactions. This phase focused on grouping customer behavior based on their transaction activities using the fuzzy c-means algorithm. In the next phase, SARDBN determined the formation of a Dynamic Bayesian network (DBN) on each cluster. Each DBN was constructed using three variables which are transaction amount, transaction mode, and period of transactions on three-time slices.

Finally, for the geographic capability aspect, Yang *et al.* in [17] proposed an AML service system for a union bank to detect money laundering on online payment using the neural network algorithm. The logical framework for this proposed method contains five sequential layers: database layer, basic data resource base layer, data analysis layer, application service layer, and the interface layer. The database layer gathers transaction information. Then, the basic data resource layer contains a knowledge base, case base, and other useful information that enabled the discovery of money laundering cases. All of the collected information then transformed into useful applications in the data analysis layer. In this layer, data cleaning is performed and then sent the result to several agents that include a neural network agent, an expert system agent, and a data mining agent to analyze. The detection component was in the application service layer where pertinent data from new incoming transactions was extricated and displayed to users to request a decision.

When money laundering is discovered, the union bank will be notified and received the result. The standard interface layer is just an interface that shows the transaction details from all financial institutions to the union bank. The main difficulty with this is the integration between different systems using different currencies. Over the past few years, several suspicious transaction detection techniques have been developed using ML techniques such as dynamic Bayesian Networks [16] and clustering [18]. Z. Chen in [11] provided an extensive survey on the applied ML techniques for anti-money laundering solutions in suspicious transaction detection depends on the AML aspects. These fuzzy logic techniques, SVM, graph clustering, frequent pattern FP close, decision tree, random tree, random forest, minimum spanning tree clustering, and genetic algorithm. On the other hand, some applications built as traditional (rule-based) expert-systems, where knowledge engineers manually extract knowledge from human experts and make it part of the inference system [19]. The performance of such expert-systems, thus, entirely depends on the quality of the acquired knowledge-base and the inference engine. These expert-systems face a significant challenge when operating in a dynamic domain as their anticipated inference capabilities are degraded with a continuously changing environment. One of the latest expert systems by Rajput in 2014 [19] used semantic web technologies to build an ontology-based expert system. The ontology consists of domain knowledge and rules which are independent from the inference system. The ontology knowledge base can be easily updated without any significant overhead and data storage requirement [19].

However, none of these mentioned above solutions covered the watch-list filtering in the AML. As discussed in the latest Chartis report, AI and ML are not yet applied on watch-list filtering in the AML solutions, because of the risk of taking a wrong decision. So in this paper, we balance between the importance of ML techniques and the industry concern about the risk of failure. Hence, we propose a model to implement ML on the watch-list filtering and introduce a new system to deploy on the financial institutes without putting them under high penalties risk.

IV. MACHINE LEARNING COMPONENT (ML-COMPONENT)

In the current implementation of AML systems, there is a significant challenge in monitoring an enormous number of transactions and evaluating the false-positive alarms. Hence, a substantial human effort is needed to do so, but not every financial institution can engage a big team of compliance officers [3]. Therefore, our objective is to develop a Machine Learning Component (dubbed as ML-Component) to predict the risky transactions before waiting in the queue for human investigation. The proposed ML-Component will be implemented as an external service (i.e., an independent component that is integrated with the watch-list filtering system through direct access to the DB) as shown in Figure 5. The DB is centralized between the SWIFT and MLComponent as clarified in the current watch-list filtering system. This will help the ML-Component to monitor coming transactions, and update the decision of the new transactions. To the best of our knowledge, this is the first implementation for ML within the watch-list filtering system. It will not be simple to take the risk of placing the ML-Component to take full control and replace human decisions. Therefore, the system will be

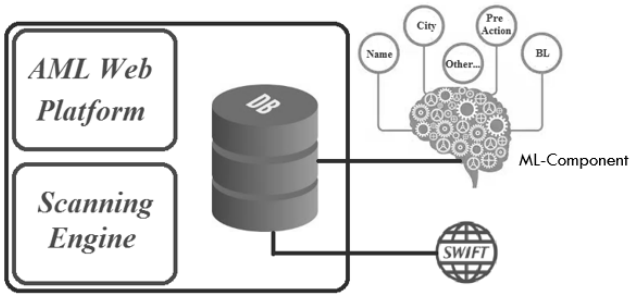


FIGURE 5. ML-Component integration with the current watch-list filtering system.

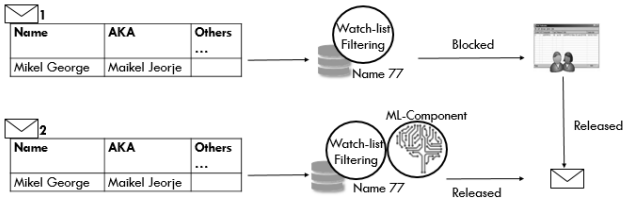


FIGURE 6. Example of the money transaction flow.

implemented gradually to help in evaluating and testing the ML-Component more efficiently. This will lead to a mature model that can be used later on with confidence.

By adding ML on the watch-list filtering system, we will use the historical data and more information about the transaction and the blacklisting entity to minimize the human effort by providing recommendations for the blocked transactions as a first step. The historical data includes all transactions evaluated with the same blacklists loaded. The ML-Component will handle the transaction flow and take an initial decision. This can be achieved by placing transactions in different queues where each queue represents the recommended decision. Later on, the ML-Component can be enhanced to replace the string matching algorithm and take a final decision for most of the transactions. However, the high-risk transactions will be investigated by a compliance officer.

Adding more rules will lead to more cases getting flagged for investigation, which will result in increasing the false-positive incidents. Hence, the ML model will be more effective than rule-based systems by improving the quality of the alerts and minimizing compliance officer costs. This will help compliance officers and investigators to focus on the most high-risk cases [20].

The following example in Figure 6 explains the message flow in the integration of ML-Component with the watch-list filtering system using a threshold rank value 75 to clarify the example. When transaction no. 1 reaches the system with no historical data and the threshold rank is 75. In this scenario, if the matching rank between the scanned financial transaction and the blacklisting entity is below the threshold value, no detection should be reported. Where if transaction no. 1 reported a hit that matched blacklisted entity information (scanned name in this example) with rank 77 (as an example depends on the value of the differences between the scanned

TABLE 2. Blocked transaction attributes (Watch-list Filtering).

No.	Attribute	Type	Accuracy Value
1	ID	Number	Unique record ID
2	Rank	Number	Matched rank
3	Amount	Number	Transferred amount
4	Birthyear	Number	Year of birth
5	Goodguy	Number	Indicates if the matched person is a GoodGuy
6	Entity type	Number	Matched entity can be person = 0 or others (company) = 1
7	Status	Number	Released = 1 or Blocked = 0

TABLE 3. Blocked transactions attributes (Swift message information).

No.	Attribute	Type	Accuracy Value
1	Source	Number	Risk rating for the source country ranged from 1 and 5.
2	Destination	Number	Risk rating for the destination country ranged from 1 and 5.

TABLE 4. Blocked transactions attributes (KYC).

No.	Attribute	Type	Accuracy Value
1	Risk Rating	Number	Clients risk rating value ranged from 1 and 5.
2	EDD/CDD	Number	Specify if Enhanced Due Diligence was done for the client.

name and the matched name on the blacklist), it will be blocked and moved to the queue waiting for the compliance officer to make a decision. If it will be released, next time a similar transaction (transaction no. 2) enters the system with the same information, the ML-Component will release it automatically.

The first step to achieve this is to prepare the dataset to be cleaned and make it compatible with the ML algorithms. Also, we have to specify the most important features. We target to test multiple algorithms to evaluate the accuracy of them and compare the results.

Our dataset contains 1500 blocked transactions with ranks between 75 and 100. The compliance officer will have limited information to decide the blocked transactions and he/she will need to check other resources. In Table 2, we explained the main features of the Watch-list Filtering system. Then, we added two other features from the swift messages which are the source country risk rate and the destination country risk rate as shown in Table 3. To increase the dataset depth in terms of useful data, we added two features from the KYC application as explained in Table 4.

Table 5 includes a sample of the dataset including ten features. The first three features are numeric, starting with the rank feature which represents the matched value with a number between 75 and 100. The second is the amount of the

TABLE 5. Sample dataset.

Rank	Amount	Birth year	Goodguy	Entity Type	Source	Destination	Risk Rating	EDDCDD	Status
79	15824	1967	1	0	1	5	3	0	1
76	591	1964	1	0	3	2	3	0	17
96	12482	1990	0	0	5	4	4	1	0

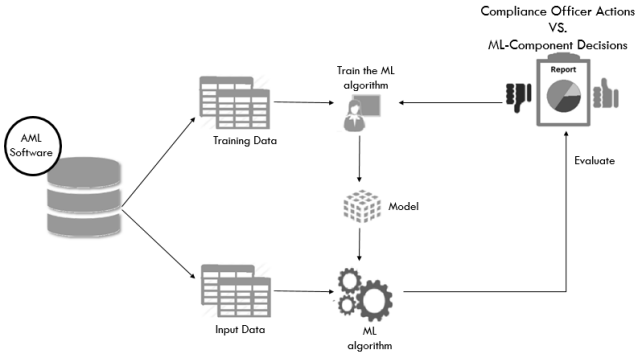


FIGURE 7. Monitoring phase.

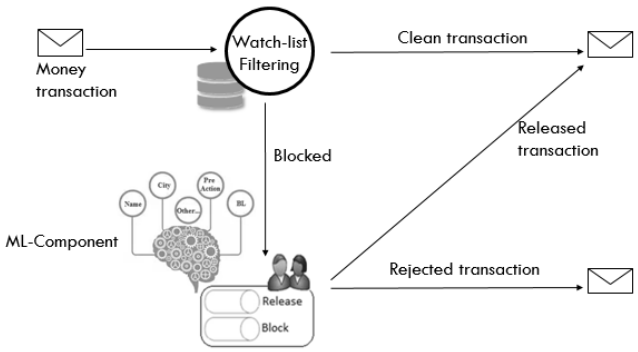


FIGURE 8. Advising phase.

transferred money. The third is the birthyear which expresses to the compliance officer the age of the sender. The fourth feature is a binary feature called goodguy, the default value is zero because all clients should be investigated by default, and one means the sender is a trusted person. The fifth feature (entity type) is binary as well, zero for individuals, and one for other types. The sixth (source), seventh (destination) and eighth (risk rating) features are categorical to show the risk rating value for both source and destination countries, and for the sender, respectively. The ninth feature is EDD/CDD, and it is a binary feature. The last one (status) is a binary feature; it is the output that we try to predict.

The proposed ML-Component will follow the following phases to implement ML on the watch-list filtering system.

A. MONITORING PHASE

The monitoring phase is the first phase of the proposed ML-Component in which the coming transactions are monitored silently. Based on the configuration of the ML-Component, it will use a portion of the transactions as training data to tune the model. Then, it will try to predict the final decision for testing data (transactions) and save it on a separate table with the transaction ID. Later on, after the compliance officer takes decisions on the same test transactions, the system will show a report that describes each transaction ID with both decisions made by the investigator and the ML-Component, as illustrated in Figure 7. The advantages of this phase can be summarized as follows.

- Check how accurate the ML-Component is: We will be able to compare the decisions taken by the ML-Component to the decisions made by the compliance officer and get a percentage of accuracy.
- Tuning the ML-Component: With the above information, we can change the included fields or their weight to a new result and compare it to the previous ones
- Convince the financial institutions to use the ML-Component: When the financial institution gets a report with exact matches between the decisions taken by the ML-Component and the decisions made by the compliance officer, they can eventually trust the ML-Component

B. ADVISING PHASE

After finishing the first phase with an acceptable report that shows the exact match between the ML-Component decision and the compliance officer decision, it is time to minimize the investigation effort by putting the ML-Component on the action. However, this will be performed gradually so that the ML-Component will not present a final decision on blocked transactions to avoid compliance risk and to monitor the system's behavior. In this phase, the ML-Component will not take full control, it will be able to make a decision, but it will not be the final decision. Human input is therefore needed to confirm the transaction decision. The watch-list filtering system can depend on the ML-Component to evaluate the pending transactions and provide the recommended decision on whether to release or reject the blocked transactions. The system will transfer the transactions to queues based on the component's recommended decision as explained in Figure 8. This will minimize the compliance officer's effort on the investigation process for blocked transactions. Furthermore, it will minimize the decision delay that may cause penalties when having a high number of pending transactions in the queue waiting for the investigation.

C. TAKEACTION PHASE

In addition to the second phase advantages, this phase will minimize the number of false-positive and false-negative transactions. The ML-Component will take a final decision

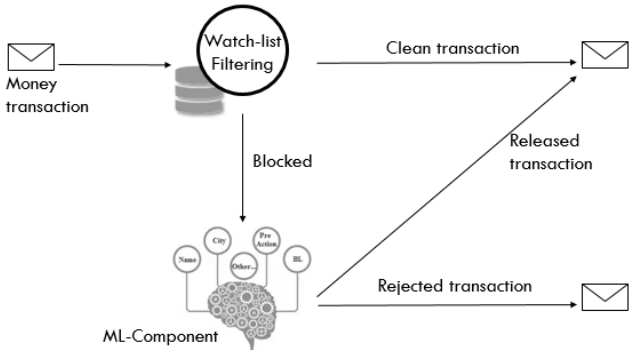


FIGURE 9. TakeAction phase.

to either release or reject the blocked transaction. This can also be configured as the compliance officer can define a set of rules where the system will behave as in the advising phase, by moving the transaction to the recommended decision queue to be processed manually. Figure 9 describes the transaction flow of the TakeAction phase.

In the watch-list filtering database, there are several useful information, either related to the transaction itself or to the blacklisted entity it matched. While pre-processing the data set and testing the model, we can select the best fields to tune the ML-Component to the best result. The financial transaction has several useful information that can be utilized to achieve the versions mentioned above and tune our ML-Component. Some of the transaction information that will be used by the ML-Component are sender reference, ordering customer, and the matched rank value.

V. PERFORMANCE EVALUATION AND RESULTS

We build multiple test scenarios to examine and compare the three algorithms (SVM, DT, and NB). The tests cover the basic features and the additional features that we added to enhance the accuracy considering the suitable normalization method. We examine two SVM test scenarios. we first used the linear default kernel, and then we changed it to poly kernel since our dataset is not linear to achieve higher accuracy. The dataset split into a training set and testing set using stratified kfold to perform multiple rounds of cross-validation with different subsets from the same data. For the continuous features, we use the Min-Max scaler to have the values of the features between 0 and 1. This will assure that the dataset is not biased to a feature than the others; which will lead to getting accurate results. A Min-Max scaler is typically done via the following equation.

$$X_normalized = \frac{(X - X_min)}{(X_max - X_min)} \quad (1)$$

Equation 1 converts the values to be between 0 and 1. X is the value to be normalized. Xmin is the minimum value for the feature and Xmax is the maximum value for the feature.

TABLE 6. Confusion matrix.

	Predicted Blocked	**Predicted Released**
Actual Blocked	True Positive Counter	False Negative Counter
Actual Released	False Positive Counter	True Negative Counter

A. PERFORMANCE METRICS

We define a set of performance metrics to evaluate the investigated ML algorithms in this thesis. We compare them to check the best algorithm to be used for high accuracy and less number of false-negative. The defined performance metrics are described below.

1) CONFUSION MATRIX

We use the confusion metrics to evaluate and compare the differences between the ML algorithms. The confusion matrix table is often used to describe the performance of a classification model on a set of test data for which the true values are known. The confusion table has the following labels as described in Table 6.

- Actual Blocked: Represents the blocked transactions on the dataset before training.
- Actual Released: Represents the released transactions on the dataset before training.
- Predicted Blocked: Represents the blocked transactions on the dataset after training.
- Predicted Released: Represents the released transactions on the dataset after training.
- True Positive Counter: Represents the number of correctly predicted positive classes.
- True Negative Counter: Represents the number of correctly predicted negative class.
- False Positive Counter: Represents the number of wrongly predicted positive classes.
- False Negative Counter: Represents the number of wrongly predicted negative class.

The true positive, true negative, false positive, and false negative are calculated by counting the transactionsṕredicted decisions compared to the transactionsóriginal decisions. The total of these counters is equal to the number of records in the dataset.

2) ACCURACY

Accuracy represents how close a measured value is to the true value. It expresses the correctness of measurement and determines by the absolute and comparative way [21]. The accuracy is calculated using Equation 2.

$$Accuracy = \frac{Sum\ of\ true\ positive + Sum\ of\ true\ negative}{Total\ population} \quad (2)$$

TABLE 7. SVM_1 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.672	0.690	0.673	0.021

TABLE 8. SVM_1 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	597	149
Actual Released	342	412

3) PRECISION

Precision metric refers to how close the measurements of the same item to each other [21]. Precision value can be calculated using Equation 3.

$$Precision = \frac{(Sum\ of\ true\ positives)}{(Sum\ of\ true\ positives + Sum\ of\ false\ positives)} \quad (3)$$

4) RECALL

This metric measures the percentage of total relevant results correctly classified by the algorithm [21]. The recall metric can be calculated using Equation 4.

$$Recall = \frac{(Sum\ of\ true\ positives)}{(Sumof\ true\ positives + Sum\ of\ false\ negative)} \quad (4)$$

B. TEST SCENARIOS

In the following, we list all tests with the details of the setup and used features. For each test, we specified the ML algorithm and listed the used features. Also, we provide the used normalization and the parameters. Then, we show the results matrices and the confusion table for each test scenario.

1) SVM_1

In this test scenario, we used the SVM algorithm with the basic features (rank, amount, birthyear, goodguy, and the entity type) as input features. For normalization, we used min-max normalization for continuous features (rank, amount, and birth year). The results are shown in Table 7 and the confusion matrix is shown in Table 8.

Table 8 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 491. 149 transactions were detected as released while they should be blocked and 342 transactions were predicted as blocked and they should be released.

2) SVM_2

In this test scenario, we used the SVM algorithm, and the used features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) in addition to two features added from swift messages (source and destination). For normalization, we used min-max normalization for continuous features (rank, amount, and birthyear). The results are shown in Table 9 and the confusion matrix is shown in Table 10.

TABLE 9. SVM_2 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.713	0.715	0.713	0.034

TABLE 10. SVM_2 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	555	191
Actual Released	239	515

Table 10 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 491. 191 transactions were detected as released while they should be blocked and 430 transactions were predicted as blocked and they should be released.

TABLE 11. SVM_3 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.815	0.816	0.816	0.028

TABLE 12. SVM_3 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	608	138
Actual Released	138	616

3) SVM_3

In this test scenario, we used the SVM algorithm, and the used features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) in addition to two features added from swift messages (source and destination) and two features added from KYC application (risk rating and cdd/edd). For normalization, we used min-max normalization for continuous features (rank, amount, and birthyear). The results are shown in Table 11 and the confusion matrix is shown in Table 12.

Table 12 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 276. 138 transactions were detected as released while they should be blocked and 138 transactions were predicted as blocked and they should be released.

4) SVM_4

In this test scenario, we used the SVM algorithm and the used features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) in addition to two features added from swift messages (source and destination) and two features

TABLE 13. SVM_4 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.821	0.824	0.821	0.020

TABLE 14. SVM_4 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	581	165
Actual Released	103	651

TABLE 15. SVM_5 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.844	0.845	0.844	0.013

TABLE 16. SVM_5 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	622	124
Actual Released	109	645

added from KYC application (risk rating and cdd/edd). For normalization, we used min-max normalization for the continuous features (rank, amount, birthyear), and the one-hot encoder for the categorical features. The results are shown in Table 13 and the confusion matrix is shown in Table 14.

Table 14 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 268. 165 transactions were detected as released while they should be blocked and 103 transactions were predicted as blocked and they should be released.

5) SVM_5

In the SVM_5 test scenario, we used the SVM algorithm. The input features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) in addition to two features added from swift messages (source and destination) and two features added from KYC application (risk rating and cdd/edd). For normalization, we used min-max normalization for continuous features (rank, amount, and birthyear). Also, we tuned the default SVM parameters and set the kernel to poly. The results are shown in Table 15 and the confusion matrix is shown in Table 16.

Table 16 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 233. 124 transactions were detected as released while they should be blocked and 109 transactions were predicted as blocked and they should be released.

6) SVM_6

In this test scenario, we used the SVM algorithm and the used features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) in addition to two features

TABLE 17. SVM_6 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.848	0.851	0.848	0.023

TABLE 18. SVM_6 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	608	138
Actual Released	89	665

TABLE 19. NB_1 rresults.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.714	0.716	0.714	0.028

TABLE 20. NB_1 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	564	182
Actual Released	247	507

added from swift messages (source and destination) and two features added from KYC application (risk rating and cdd/edd. For normalization, we used min-max normalization for the continuous features (rank, amount, birthyear), and the one-hot encoder for the categorical features. Also, we tuned the default SVM parameters and set the kernel to poly. The results are shown in Table 17 and the confusion matrix is shown in Table 18.

Table 18 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 227. 138 transactions were detected as released while they should be blocked and 89 transactions were predicted as blocked and they should be released.

7) NB_1

In this test scenario, we used the NB algorithm with the used features are the basic ones (rank, amount, birthyear, goodguy and the entity type) as input features. For normalization, we used min-max normalization for continuous features (rank, amount, and birthyear). The results are shown in Table 19 and the confusion matrix is shown in Table 20.

Table 20 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 429. 182 transactions were detected as released while they should be blocked and 247 transactions were predicted as blocked and they should be released.

8) NB_2

In this test scenario, we used the NB algorithm and the used features are the basic ones (rank, amount, birthyear, goodguy,

TABLE 21. NB_2 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.709	0.711	0.709	0.027

TABLE 22. NB_2 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	561	185
Actual Released	251	503

TABLE 23. NB_3 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.806	0.807	0.806	0.016

TABLE 24. NB_3 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	604	142
Actual Released	149	605

and the entity type) in addition to two features added from swift messages (source and destination) as input features. For normalization, we used min-max normalization for continuous features (rank, amount, birthyear). The results are shown in Table 21 and the confusion matrix is shown in Table 22.

Table 22 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 436. 185 transactions were detected as released while they should be blocked and 251 transactions were predicted as blocked and they should be released.

9) NB_3

In this test scenario, we used the NB algorithm and the used features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) in addition to two features added from swift messages (source and destination) and two features added from KYC application (risk rating and cddedd. For normalization, we used min-max normalization for continuous features (rank, amount, and birthyear). The results are shown in Table 23 and the confusion matrix is shown in Table 24.

Table 24 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 291. 142 transactions were detected as released while they should be blocked and 149 transactions were predicted as blocked and they should be released.

10) NB_4

In this test scenario, we used the NB algorithm and the used features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) in addition to two features

TABLE 25. NB_4 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.824	0.827	0.824	0.019

TABLE 26. NB_4 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	586	160
Actual Released	103	651

TABLE 27. DT_1 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.637	0.638	0.637	0.038

TABLE 28. DT_1 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	478	268
Actual Released	276	478

added from swift messages (source and destination) and two features added from KYC application (risk rating and cdd/edd) as input features. For normalization, we used min-max normalization for the continuous features (rank, amount, and birthyear), and the one-hot encoder for the categorical features. The results are shown in Table 25 and the confusion matrix is shown in Table 26.

Table 26 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 263. 160 transactions were detected as released while they should be blocked and 103 transactions were predicted as blocked and they should be released.

11) DT_1

In this test scenario, we used the DT algorithm and the used features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) as input features. For normalization, we used min-max normalization for continuous features (rank, amount, and birthyear). The results are shown in Table 27 and the confusion matrix is shown in Table 28.

Table 28 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 544. 268 transactions were detected as released while they should be blocked and 276 transactions were predicted as blocked and they should be released.

12) DT_2

In this test scenario, we used the NB algorithm with the used features are the basic ones (rank, amount, birthyear,

TABLE 29. DT_2 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.668	0.670	0.668	0.047

TABLE 30. DT_2 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	486	260
Actual Released	237	517

TABLE 31. DT_3 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.782	0.784	0.782	0.037

TABLE 32. DT_3 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	586	160
Actual Released	166	588

goodguy, and entity type) in addition to two features added from swift messages (source and destination) as input features. For normalization, we used min-max normalization for continuous features (rank, amount, and birthyear). The results are shown in Table 29 and the confusion matrix is shown in Table 30.

Table 30 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 497. 260 transactions were detected as released while they should be blocked and 237 transactions were predicted as blocked and they should be released.

13) DT_3

In this test scenario, we used the DT algorithm with the used features are the basic ones (rank, amount, birthyear, goodguy, and entity type) in addition to two features added from swift messages (source and destination) and two features added from KYC application (risk rating and cdd/edd) as input features. For normalization, we used min-max normalization for continuous features (rank, amount, and birthyear). The results are shown in Table 31 and the confusion matrix is shown in Table 32.

Table 32 shows the number of transactions which detected with the wrong decision comparing to the real transactions with a total of 326. 160 transactions were detected as released while they should be blocked and 166 transactions were predicted as blocked and they should be released.

TABLE 33. DT_4 results.

Accuracy Average	Precision Average	Recall Average	Accuracy Standard Deviation
0.783	0.786	0.783	0.027

TABLE 34. DT_4 confusion matrix.

	Predicted Blocked	Predicted Released
Actual Blocked	579	167
Actual Released	158	596

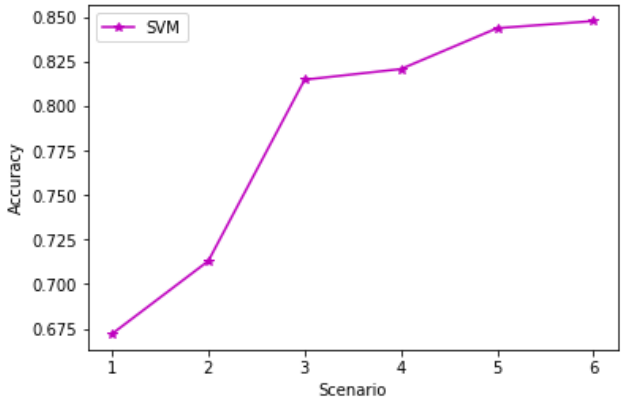


FIGURE 10. SVM test scenarios accuracy.

14) DT_4

In this test scenario, we used the DT algorithm with the used features are the basic ones (rank, amount, birthyear, goodguy, and the entity type) in addition to two features added from swift messages (source and destination) and two features added from KYC application (risk rating and cdd/edd) as input features. For normalization, we used min-max normalization for the continuous features (rank, amount, and birthyear), and the one-hot encoder for the categorical features. The results are shown in Table 33 and the confusion matrix is shown in Table 34.

Table 34 shows the number of transactions that detected with the wrong decision comparing to the real transactions with a total of 325. 167 transactions were detected as released while they should be blocked and 158 transactions were predicted as blocked and they should be released.

C. EXPERIMENTS SUMMARY AND ASSESSMENTS

In Figure 10, we compare the tests' results for all scenarios using SVM considering different settings. The graph shows the enhancement done on accuracy. The accuracy was around %67 in test one and we managed to enhance it to around %85 in test 6 after using the additional features from swift messages and KYC application and did the normalization for all features.

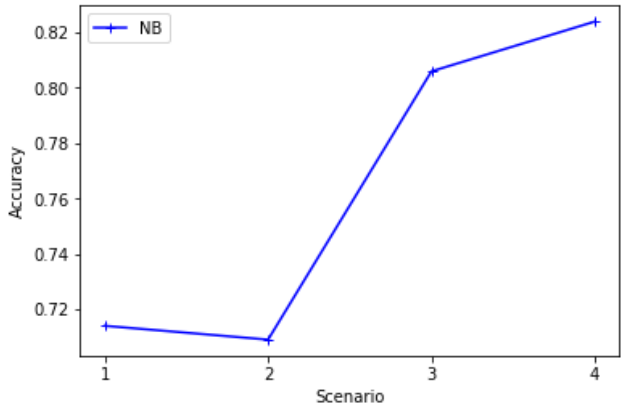


FIGURE 11. NB test scenarios accuracy.

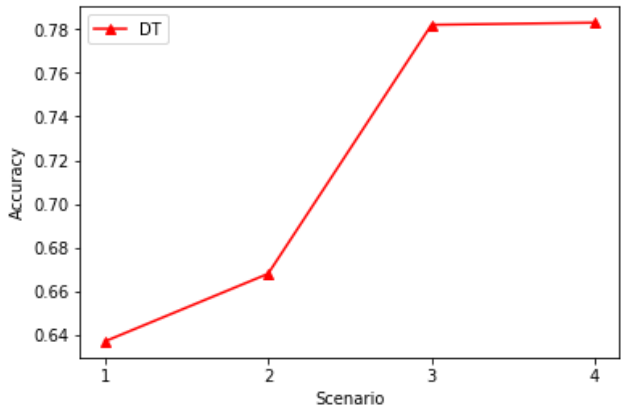


FIGURE 12. DT test scenarios.

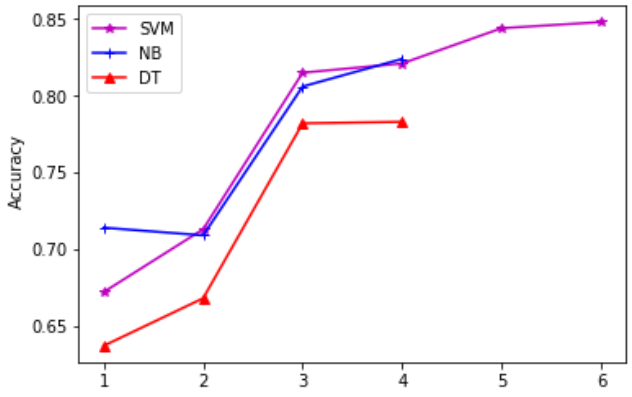


FIGURE 13. ML algorithms comparison.

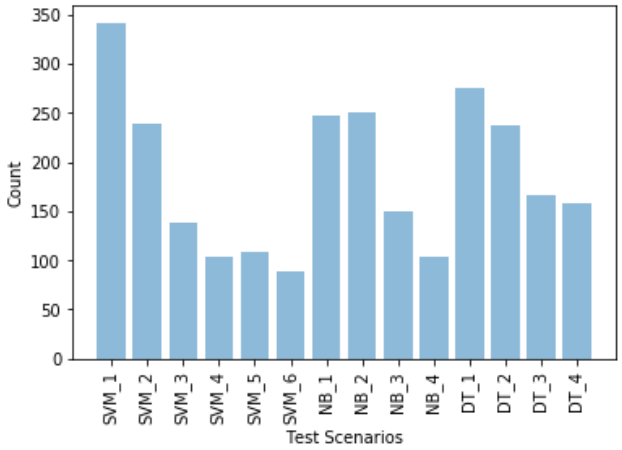


FIGURE 14. Number of false-positive.

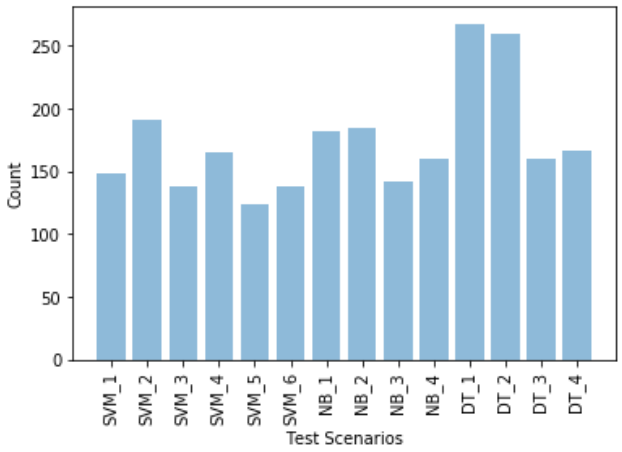


FIGURE 15. Number of false-negative.

While in Figure 11, we enhanced the NB algorithm's accuracy from around 72% to around 82% by including the additional features and normalizing the features.

Figure 12 shows the DT enhancement we achieved from 64% to 78% for the accuracy even we tried all the enhancement parameters used on the SVM and NB algorithms.

We compared all algorithms in Figure 13 to clarify the differences between the three algorithms and the enhancement achieved using the SVM algorithm. Based on that, the SVM achieves higher accuracy using the available features and both min-max and one hot encoder normalization methods. Also, we achieve the lower false-positive and false-negative counters using the SVM_6.

Besides the accuracy, we have two major elements we need to consider while evaluating the efficiency of the ML algorithm; the False- Positive and the False-Negative. The False- Positive counter reflects the number of transactions that predicted as released transactions while the actual status was rejected. The False-negative counter reflects the number of transactions that predicted as rejected transactions while the actual status was released. It is good to minimize both values but the False Positive is riskier because releasing a risky transaction will put the financial institution at risk. Figure 14 shows that the test scenario SVM_6 achieved the minimum value of the False-Positive. This will minimize the compliance officer investigation effort.

Figure 15 shows the test scenario SVM_5 achieved the minimum value of False Negative. And the second minimum is the test scenario SVM_6. Because we care more about the

False-positive accuracy, we consider the test scenario SVM_6 as the best enhancement we can achieve by applying ML to the watch-list filtering system.

VI. CONCLUSION AND FUTURE WORK

Financial institutions are considered part of the front line to fight money laundering and terrorist financing. Also, the financial institutions need to accelerate the investigation process to improve the ''time to value,'' which is the required time to finish a transaction life-cycle. So by applying ML on the watch-list filtering applications that monitor the financial transactions is a must to fight financial crime with better performance and shorter time. Many work and investigations have been done to apply ML algorithms on AML solutions, but the industry has concerns related to automating regulatory compliance areas, because of the high penalties if any failure happens. So, up to the best of our knowledge, this paper is the first research work that introduces a way to automate the blocked transactions process through three phases.

In this paper, we summarized the main applications of the ML algorithms on the AML solutions, the KYC mainly and its six aspects, then we introduced a high-level architecture for building and integrating the ML-Component with watch-list filtering AML system. A full study has been done to be able to avoid the above concerns and the industry worries by planning the implementations on multiple phases. We performed multiple experiments using different settings on ML algorithms (SVM, DT, and NB), and we found that the SVM outperforms other algorithms. Because our dataset is nonlinear, we used the polynomial kernel (mathematical function) and achieved higher accuracy for predicting the transactionś decision. We used the correlation matrix to show the relationship between the numeric features.

In the future, we will focus on implementing the first version to apply ML on watch-list filtering. Work will be divided into small parts, each of which can build and achieve specific functionality. This will make it easier to validate the result and evaluate the work on both sides, business, and technology. The development for the ML-Component will go through three phases, starting by building and tuning the component, then using the component decision as advice to the compliance officer. Then, it can take a final decision for certain confident cases. So, as mentioned earlier, it is risky to entirely depend on the intelligent component for controlling the financial transactions, taking into consideration the industry concerns for automated regulatory compliance areas, we will integrate the component efficiently with current AML applications.

REFERENCES

[1] (2012). *Study Guide for The CAMS (Certified Anti-Money Laundering Specialists)*. [Online]. Available: https://www.acams.org/

[2] R. J. Lowe, ''Anti-money laundering—The need for intelligence,'' *J. Financial Crime*, vol. 24, no. 3, pp. 472–479, Jul. 2017.

[3] M. Hinkle and D. Stewart. (2014). *Modernizing Anti-Money Laundering Practices*. [Online]. Available: https://www.sas.com/content/dam/SAS/en_us/doc/conclusionpaper1/modernizingamlpractices-106930.pdf

[4] C. Gatti and J. Eligon. (Dec. 2009). *Iranian Dealings Lead to a Fine for Credit Suisse*. [Online]. Available: https://www.nytimes.com/2009/12/16/business/16bank.html/nytimes.co

[5] Chartis. (2017). *RiskTech100 RiskTech100 2018*. [Online]. Available: https://www.risktech100.com/2018-report

[6] R. Ramachandran. (2014). *OFAC Name Matching and False Positive Reduction Techniques*. [Online]. Available: https://www.cognizant.com/InsightsWhitepapers/OFAC-Name-Matching-and-False-Positive-Reduction-Techniques-codex1016.pdf

[7] Standards MT. (Nov. 2019). *Standards Release Guide*. [Online]. Available: https://www.swift.com/standards

[8] A. Lait and B. Randell, ''An assessment of name matching algorithms,'' Series-Univ., Tech. Rep., 2006. [Online]. Available: https://www.semanticscholar.org/paper/An-Assessment-of-Name-Matching-Algorithms-Lait-Randell/bd88761329102ea617c1c3173cf11efac4ae7876

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

[10] Chartis. (2019). *Artificial Intelligence in Financial Services, 2019: Demand-side Analysis*. [Online]. Available: https://www.chartis-research.com/technology/artificial-intelligence-ai/artificial-intelligence-financial-services-2019-demand-side-analysis-10716

[11] Z. Chen, L. D. Van Khoa, E. N. Teoh, A. Nazir, E. K. Karuppiah, and K. S. Lam, ''Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: A review,'' *Knowl. Inf. Syst.*, vol. 57, no. 2, pp. 245–285, Nov. 2018.

[12] S. Gao, D. Xu, H. Wang, and P. Green, ''Knowledge-based anti-money laundering: A software agent bank application,'' *J. Knowl. Manage.*, vol. 13, no. 2, pp. 63–75, Apr. 2009.

[13] D. Cao and P. Do, ''Applying data mining in money laundering detection for the vietnamese banking industry,'' in *Proc. 4th Asian Conf. Intell. Inf. Database Syst. (ACIIDS)*, Kaohsiung, Taiwan, Mar. 2012, pp. 207–216.

[14] R. Dreáewski, J. Sepielak, and W. Filipkowski, ''The application of social network analysis algorithms in a system supporting money laundering detection,'' *Inf. Sci.*, vol. 295, pp. 18–32, Feb. 2015.

[15] K. Chitra and B. Subashini, ''Data mining techniques and its applications in banking sector,'' *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 8, pp. 219–226, Aug. 2013.

[16] S. Raza and S. Haider, ''Suspicious activity reporting using dynamic Bayesian networks,'' in *Proc. World Conf. Inf. Technol.*, Feb. 2011, pp. 987–991.

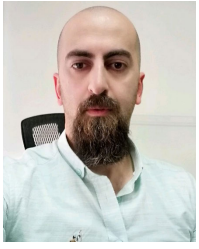
[17] Q. Yang, B. Feng, and P. Song, ''Study on anti-money laundering service system of online payment based on union-bank mode,'' in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput. (WiMob)*, Marrakesh, Morocco, Sep. 2007, pp. 4991–4994.

[18] A. Larik and S. Haider, ''Clustering based anomalous transaction reporting,'' in *Proc. World Conf. Inf. Technol.*, Feb. 2011, pp. 606–610.

[19] Q. Rajput, N. S. Khan, A. Larik, and S. Haider, ''Ontology based expert-system for suspicious transactions detection,'' *Comput. Inf. Sci.*, vol. 7, no. 1, pp. 103–114, Jan. 2014.

[20] T. Horan, F. Holzenthal, and S. Zoldi, ''Advancing AML compliance with artificial intelligence,'' FICO, Bengaluru, Karnataka, Tech. Rep., 2017. [Online]. Available: https://www.fico.com/en/latest-thinking/white-paper/advancing-aml-compliance-artificial-intelligence

[21] M. Arunadevi and J. Nithya, ''Comparison of feature selection strategies for classification using rapid miner,'' *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 4, pp. 13556–13563, 2016.



MOHANNAD ALKHALILI received the B.Sc. and M.Sc. degrees in computer science from Applied Science Private University, Amman, Jordan, in 2004 and 2020, respectively. He has more than 16 years of experience in software development. He spent nine years working in building Anti-money Laundering applications for money transactions monitoring.

MAHMOUD H. QUTQUT (Member, IEEE) received the B.Sc. degree in computer systems from Applied Science University, Jordan, in 2004, the M.Sc. degree in telecommunication systems from DePaul University at Chicago, Illinois, in 2009, and the Ph.D. degree from Queen's University, Canada, in 2014, where he worked under the supervision of Professor Hossam S. Hassanien. He is currently an Associate Professor with the Faculty of Information Technology, Applied Science University, Amman, Jordan, since October 2014. He was a Visiting Assistant Professor with the Queen's University School of Computing, from March 2017 till Sept 2019, and he is a Research Associate with the Queen's Telecommunications Research Lab. He is a member of the ACM, and has served as Technical Program Committee co-chair, member and reviewer of many international conferences and journals. He has several publications of the top tier venues in the research area of networking like IEEE ICC and IEEE Globecom.

FADI ALMASALHA (Member, IEEE) received the M.Sc. degree in computer science from the New York Institute of Technology, in 2005, and the Ph.D. degree in computer science from the University of Illinois at Chicago, in 2011. He is currently an Associate Professor with the Faculty of Information Technology, Applied Science Private University, Amman, Jordan. In fall of 2011, he joined the Department of Computer Science at the Applied Science University. He received his Associate rank on 2016, during his appointment as the head of computer science department. He has published more than ten technical papers, journals, and book chapters in refereed conferences and journals in the areas of multimedia systems, data mining, and cryptography.