Computers and Electrical Engineering 93 (2021) 107215

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Big Data Analytics-as-a-Service: Bridging the gap between security experts and data scientists☆

Claudio A. Ardagna [a], Valerio Bellandi [a,*], Ernesto Damiani [a], Michele Bezzi [b], Cedric Hebert [b]

[a] *Computer Science Department, Università Degli Studi di Milano, Via Celoria 18, Milan, Italy*
[b] *SAP Security Research, Avenue du Dr Maurice Donat 805, Mougins, France*

ARTICLE INFO

Keywords:
Artificial intelligence
Big Data Analytics
Machine learning
Security and privacy

ABSTRACT

We live in an interconnected and pervasive world where huge amount of data are collected every second. Fully exploiting data through advanced analytics, machine learning and artificial intelligence, becomes crucial for businesses, from micro to large enterprises, resulting in a key advantage (or shortcoming) in the global market competition, as well as in a strong market driver for business analytics solutions. This scenario is deeply changing the security landscape, introducing new risks and threats that affect security and privacy of systems, on one side, and safety of users, on the other side. Many domains that can benefit from novel solutions based on data analytics have stringent security requirements to fulfill. The Energy domain's Smart Grid is a major example of systems at the crossroads of security and data-driven intelligence. The Smart Grid plays a crucial role in modern energy infrastructure. However, it must face two major challenges related to security: managing front-end intelligent devices such as power assets and smart meters securely, and protecting the huge amount of data received from these devices. Starting from these considerations, setting up proper analytics is a complex problem because security controls could have the undesired side effect of decreasing the accuracy of the analytics themselves. This is even more critical when the configuration of security controls is let to the security expert, who often has only basic skills in data science. In this paper, we propose a solution based on the concept of Model-Based Big Data Analytics-as-a-Service (MBDAaaS) that bridges the gap between security experts and data scientists. Our solution acts as a *middleware* allowing a security expert and a data scientist to collaborate to the deployment of an analytics addressing their needs.

1. Introduction

Despite the success of Big Data Analytics (BDA), there are a series of challenges that need to be addressed for fully leveraging their potential in the plethora of different scenarios that characterize the current business world. Starting from data integration and data governance, Big Data are characterized by large diversity and dynamics, as epitomized by the 5 V storyline (Volume, Velocity, Veracity, Variability, Value), calling for data preparation, pipelining, and orchestration solutions that are easy to build and adapt. Moreover, once the data issues are solved, similar requirements apply to the development and deployment of the corresponding analytics. The latter should address business needs, being accessible to a business user, and be flexible and adaptive to run multiple

☆ This paper is for special section VSI-gridl. Reviews processed and recommended for publication by Guest Editor Dr. G. Jeon.

* Corresponding author.

E-mail address: valerio.bellandi@unimi.it (V. Bellandi).

https://doi.org/10.1016/j.compeleceng.2021.107215
Received 29 July 2020; Received in revised form 29 March 2021; Accepted 11 May 2021
Available online 17 May 2021

"*what if*" scenarios where several models must be employed and retrained as data changes. Lastly, BDA needs to be integrated with convenient data visualization tools. While a lot of effort has been devoted to the above issues [1], fulfilling non-functional requirements is still more an art than a science. By increasing the data volume, the performance should be optimized, for instance, to dynamically allocate additional resources without disrupting the analytics pipeline. Security, privacy compliance requirements are equally important and present critical challenges related to data volume as well.[1]

For small businesses or companies operating in niche business scenarios, the capability of building analytical models in days based on business level descriptions, without the need for additional coding, is priceless. On the other hand, automated deployment of data analytics pipelines on the part of non-computer-savvy users may open the door to new security threats, plaguing the operation of the systems and the safety of their users. Critical infrastructures receive sophisticated and targeted attacks (e.g., ramsonware, mobile malware) every day, while 43% of cyber attacks target small businesses, with a total estimated loss of around 45 billion dollars in 2018 [3]. We argue that security experts are not enough for security management; they in fact need to team up with data scientist designing analytics applicable to large amounts of (possibly sensitive) data with privacy requirements in mind. Combining both types of expertise in a single individual is unlikely, and there is the need of providing technical solutions to bridge the gap between data scientists and security experts.

This paper aims to fill in this gap by providing an environment where security experts and data scientists can collaborate to implement a proper data analytics process for cybersecurity. The proposed approach is based on a methodology and framework for Model-based Big Data Analytics-as-a-Service (MBDAaaS) [4], which, on one side, helps security experts in preparing and deploying a data analytics that addresses their requirements and, on the other side, provides full transparency on execution workflows and computations (Section 2). MBDAaaS implements a multi-step process as follows. First, the security expert and the data scientist collaborate to the definition of a *declarative model*, specifying the requirements (from security to privacy) of a given data analytics, in the form of goals describing the aim of the analytics and features to be supported by the execution environment. Second, the data scientist uses the requirements in the declarative model to generate a platform-independent *procedural model* specifying how analytics are carried out and composed in terms of abstract services. Finally, the data scientist uses smart engines to compile the procedural model in a ready-to-be-executed *deployment model* specifying Big Data platform-dependent configurations and supporting automatic provisioning of computational components and resources. This co-operation based on MBDAaaS provides short roll-out time, supports a quick trial-and-error problem solving, and fosters reuse.

The paper focuses on the energy domain and Smart Grids, where the monitoring of energy usage is done in real time by means of smart meters' bidirectional communication [5]. The power supply system and its protection devices are monitored by control centers for securing the Smart Grid's load balancing system during communication. Cloud Computing is used to support communication between substations and the power supply companies' power plants. Built-in redundancy is utilized to increase the reliability, security and robustness of this communication [6]. Over the time of the day, energy demand varies as the utilization differentiates between day (peak operation) and night (lower level operation). The need of preserving the security and privacy of consumer data is critical to the acceptance of Smart Grids, especially when cloud service providers are involved. In other words, utilities are reluctant to face privacy issues concerning their customers' data [7].

Our methodology and its relevance from a business point of view are discussed and validated in the context of privacy-aware analytics for security incidents and malware detection (Sections 3 and 4). Our real-world reference scenario is a privacy-preserving log analysis for the detection of security incidents, proposed by SAP, one of the largest Software and Information Technology services groups worldwide. This applies to several critical processes within Smart Grid management, and in particular to the billing application, a major target for disclosure attacks [8]. Examples of attacks targeting a billing application are: *(i)* exfiltration of sensitive data contained in the billing database, *(ii)* escalation of privileges within the application itself, allowing a user with read-only privileges to access parts of the data and obtain write access, or even administrator accesses for the application. Detecting security incidents at the application level raises a number of concerns. The first one is that each billing application is different and has its own log mechanism, introducing the need of specific detection rules. In addition the log itself is sensitive and typically cannot be shared, posing a challenge that the people aware of the specific business and application risks do not have access to the data they need for defining and tuning detection rules. In the remaining of this paper, we present how MBDAaaS can be used to engage security experts and data scientists to monitor the security of a billing application in Smart Grid with privacy in mind, and compare it against similar solutions (Section 5).

2. Data analytics pipeline

Many times, data scientists have been put under the spotlight as the – supposedly – protagonists of the Big Data revolution in companies [9]. Firms need to get the right analytical skills and expertise added to their human capital, but this goes well beyond hiring data scientists. Managers still question themselves on which new talent they need and how to upgrade the skills of their human resources. Demauro et al. [10] provided a landscape of Big Data-related human-resource needs, offering a systematized nomenclature of job roles and skills. They proposed a semi-automated analytical process, based on web scraping, expert judgment, text mining, and topic modeling techniques, to review job offers related to Big Data, using more than 2.700 job descriptions posted online. Their findings confirmed the ideas in [11], suggesting that data scientists alone are far from being sufficient in granting companies a real

[1] According to [2], 90% of all existing data has been collected in 2 years at a rate of 2.5 quintillion bytes per day with an increasing trend.

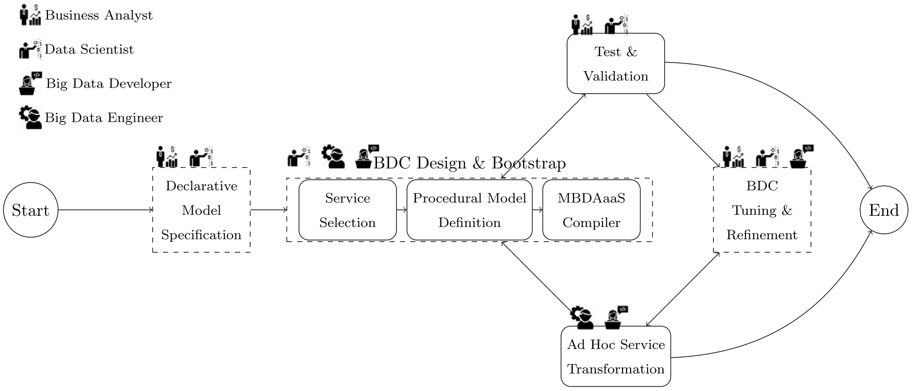


Fig. 1. MBDAaaS Middleware.

competitive advantage. From this analysis, the authors proposed 4 different job families related to Big Data campaigns: *(i)* Business Analysts, *(ii)* Data Scientists, *(iii)* Big Data Developers, and *(iv)* Big Data Engineers.[2]

Starting from these considerations, a MBDAaaS methodology [4,12], driving automatic setup and execution of data analytics, can suggest a different mix of job families. In [13], we proposed a new Development Life Cycle (DLC) for MBDAaaS, supporting fast comparison of alternative solutions and incremental refinement and optimization of such solutions. In this paper, we adopt the MBDAaaS approach in Fig. 1 as a middleware orchestrating the actors of a generic business process to the aim of implementing a proper Big Data analytics. It bridges the gap between business analysts, data scientists, Big Data developers, and Big Data engineers in the design, execution, and maintenance of complex and continuous data analytics pipelines. This solution organizes the MBDAaaS process into three phases (dashed rectangles in Fig. 1) and two decision points (rounded rectangles in Fig. 1) as follows.

Declarative Model Specification. Declarative model specification is the first step of MBDAaaS [12]. The Big Data *customer* produces a model specifying the requirements of a Big Data Campaign (BDC). Business Analysts and Data Scientists collaborate to complete this model without considering the technical aspects.

BDC Design and Bootstrap. Triggered by a declarative model specification, the second step of MBDAaaS [12] results in the definition of a ready-to-be executed deployment model. During this phase, Big Data Developers and Engineers work in conjunction with Data Scientists. Services compatible with the declarative model specification are first selected (Service Selection Fig. 1). Compatible services are made available through a service catalog, providing services at two layers of abstraction: *(i)* abstract services that can be used to define the procedural model and only consists of service APIs; *(ii)* service instances that are those services available in the target execution platform and are linked back to the corresponding abstract service. Each abstract service is annotated with a Boolean expression of declarative requirements, mapping them to objectives in the declarative model. These annotations form a database that can be queried to select a set of services compatible with the declarative model objectives [12]. The Big Data *consultant* uses the MBDAaaS platform to connect these services, obtaining an abstract workflow of BDC. The generated workflow, jointly defined by Big Data Developers and Engineers, represents the procedural model (Workflow Definition in Fig. 1). Starting from the procedural model, our MBDAaaS platform generates a platform-dependent workflow, called deployment model (MBDAaaS Compiler in Fig. 1).

Test and Validation. Upon retrieving the deployment model, MBDAaaS platform executes the analytics on the target Big Data platform. The result of the execution of a service composition is evaluated at *Test and Validation* decision point. Two possible alternatives can be followed after evaluation: *(i)* the service composition must be redesigned by substituting (BDC Design and Bootstrap) or refining (BDC Tuning and Refinement) one or more services, *(ii)* the desired quality has been achieved and the service composition can be deployed in production (End).

BDC Tuning and Refinement. The user refines a (subset of) service by developing her own algorithms, either starting from an existing sequential – platform independent – code or completely from scratch.

Ad Hoc Service Transformation. The result of the execution of the ad hoc computation is evaluated at *Ad Hoc Service Transformation* decision point. Three possible alternatives can be followed after the evaluation: *(i)* the service composition must be redesigned (BDC Design and Bootstrap), *(ii)* additional tuning or refinement is required (BDC Tuning and Refinement), or *(iii)* the desired quality is achieved and the code-based, multi-platform implementation can be exposed as a service and included in the catalog (End). We note that the developed components can be included in the catalog to enrich phase design and bootstrap, and incrementally reduce the need for ad hoc coding and refinement.

[2] We note that the security expert in our paper resembles to business analyst and big data developer in [10].

3. Privacy-preserving log analyzer

We show how MBDAaaS can be used to manage the requirements of a data-intensive security application. We consider a Threat Detection System (TDS) for a cloud application provider whose non-functional characteristics are particularly relevant for our scenario. Modern TDSs [14] evolved from the network level and recently started encompassing the application layer. TDSs *(i)* must involve two different, and high-skilled, professional profiles: data scientist and security expert; *(ii)* should be adaptive and need multiple "trial-and-error" development cycles to address the evolving threat landscape nature; *(iii)* must be modular to handle the huge amount of data (e.g., log files) generated by large cloud infrastructures. TDSs detect attacks by gathering and analyzing log data, such as user change logs, security audit logs, remote function call gateway logs, and transaction logs. Logs are pre-processed and analyzed using security analytics, such as anomaly detection algorithms, which can highlight suspicious events. On top of the generated events and alerts, a detailed investigation is performed by a security expert in the security operation team, to decide whether a real attack was detected or was a false positive. With the increasing volume and diversity of log data, and continuous emergence of new attacks, the data pipeline and the specific analytics have to be often adapted and redesigned. An additional hurdle is that log files may contain personal information (e.g., user IDs, IP addresses), which are subject to internal policies and regulatory constraints, and, beyond the authorized personnel based on the "need to know" principle (e.g., security operation team), other roles (e.g., developers or data scientists) may have a limited access to them. When searching for security incidents, analytics need to detect either a deviation from a standard behavior (unplanned anomalous activity), during or outside of an exceptional process (planned anomalous activity), or regular malicious activities merged into the normal state of operations (unplanned ordinary activities such as advanced persistent threats or repeated frauds).

Advances in ICT contributed to the transformation of the traditional electricity grid into the smart grid. Smart grid cyber-security issues include ensuring the Confidentiality, Integrity, and Availability (CIA) triad of the control systems and ICT. CIA triad is essential for both communication infrastructures and the protection, operation, and management of the energy [15]. Starting from these considerations, we describe the design of a flexible BDA pipeline supporting: *(i)* provisioning of customized analytics and reporting, *(ii)* data anonymization (at different levels, for different users, for different purposes), *(iii)* scalability to variable data loads, *(iv)* semi-automatic integration of new logs and analytics.

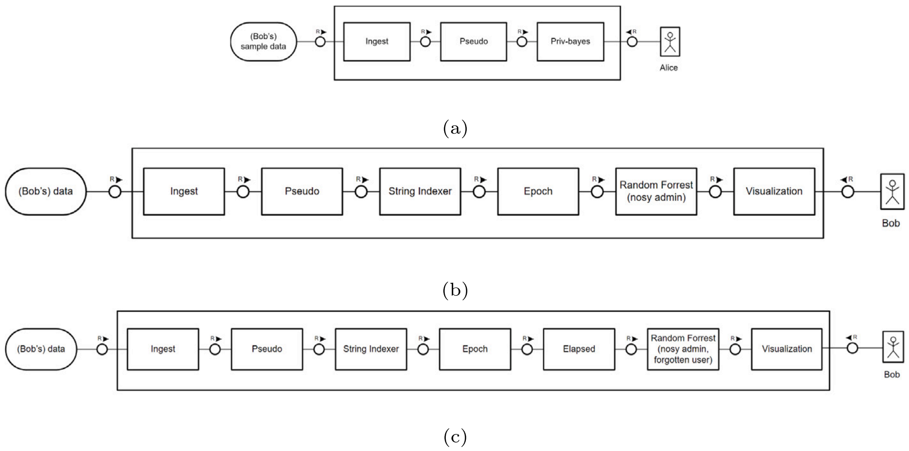
Starting from an organizational perspective, billing application logs are processed and analyzed by *security operation teams*, who have clearance to access and process confidential information. They rely on TDS tools based on Machine Learning models. However, specialized data science skills are needed for devising and training such models. Ideally, we would employ human resources having deep expertise both in security and data science, but they are rare. Actually, there is a shortage of security experts and data scientists, and the trend is increasing [16]. Hiring or upskilling talents in both fields for applying machine learning to cybersecurity is not a viable and scalable approach. An alternative is to keep roles separated, which brings up two issues:

- the data scientist/technologist know how to devise a machine learning pipeline, from data preparation to deployment, but they are not authorized to access the data [17];
- the security operation team is legitimated to access the data and can define the requirements for identifying incidents, but cannot cope with the complexity of setting up the machine learning tools.

MBDAaaS bridges the gap between these two roles and lets them collaborate. Let us consider the target goal of running an analytics on the grid's billing application logs, to automatically classify certain behaviors as security incidents. More specifically, let us focus on *privilege abuse* attacks on the part of database administrators who can access all content, including sensitive data. Database administrators are chosen carefully and sign a document upon taking up the role, stating that they will use their account only for performing administrative tasks such as data backups, adding/suppressing users, or modifying user authorizations. Technically, nothing prevents an administrator to read sensitive database tables, but each read request is being logged. It is not possible to deduce from the log whether the currently logged-in user is an administrator or not, but it is possible to mark as malicious those activities where users, usually performing administrative actions, execute a read action on a sensitive table. This incident is usually reported as an occurrence of a *nosy admin*. Let us assume that we want to build a classifier for detecting *nosy admin* activities from logs. At the same time, let us also consider an extension of the previous scenario to detect *dormant accounts*. A dormant account is an account, which has not been used for a long time and suddenly becomes active again. This can happen for a number of reasons, the most common one is an employee, who leaves the company, but whose account is not deactivated in the system. If the former employee decides to connect or get his account hacked through a password guessing attack, the account can be used for ex-filtrating data or performing other attacks. The extension is related to the original scenario, as nosy admins acting out of dormant accounts are particularly suspicious.

This twofold analysis needs to be run by the operation team because it processes confidential data; but it needs to be built by data scientists/technologists capable to design the analytics. To implement this scenario, we use three MBDAaaS pipelines (Fig. 2) as follows:

- **Bootstrap**: this pipeline anonymizes training data, supporting the development of a classifier service (Fig. 2(a)).
- **Detection of *nosy admin***: this pipeline works in production and, once deployed, detects (classifies) *nosy admin* attack (Fig. 2(b)).
- **Detection of *dormant account***: this pipeline adapts the production *nosy admin* pipeline for detecting dormant accounts (Fig. 2(c)).



(a)

(b)

(c)

Fig. 2. MBDAaaS Pipeline: *(a)* Bootstrapping pipeline, (b) Detection of *nosy admin* activity pipeline, (c) Detection of *dormant account* pipeline.

3.1. Bootstrap pipeline: Data anonymization

This pipeline anonymizes data before using them in the training process to build the classifier. Following our methodology (Section 2), the security expert in the operation team (henceforth called Bob) defines the high-level requirements of the solution in terms of data preparation (ingestion) and anonymization, releasing the declarative model for the pipeline. After defining the declarative model and checking its consistency against conflicting requirements via the MBDAaaS platform [12], the compatible services are retrieved from the service catalog. Following the process presented in Fig. 1, the deployment model realizing the pipeline (Fig. 2(a)) for anoymizing data is generated and executed. In general, two issues can arise.

- One or more services are not available in the catalog. This is quite likely for the data preparation/ingestion service, since there is a huge variety of log-types and different logs should be harmonized to a common format. In this case, Bob can *(i)* export a very small subset of its log data, checking they do not contain sensitive information, or *(ii)* create a few artificial entries respecting the data structure of the original data and then provide them to the data scientist (henceforth called Alice) to define the necessary ingestion service.
- The choice of specific anonymization algorithms could be beyond the skills of Bob and, already in this phase, he may need the support of Alice, who can suggest, for example, a combination of standard pseudo-anonymization services with more sophisticated ones, such as Priv-Bayes [18] (see Fig. 2(a)).

Once the bootstrap pipeline is successfully deployed, Bob can feed it with operational data. Alice has now access to a high enough amount of data for training and tuning Machine Learning models, such as a Random Forest classifier. During this phase, Alice may choose to add some new services, for example for data preparation, like converting categorical in numerical (*String Indexer*) data, or representing dates in seconds (*Epochs*). These services are then added to the catalog, and their links with the classifier are added to the “knowledge” of the MBDAaaS platform for realizing the correct service composition [12].

Once the Random Forest classifier is tuned to an acceptable level of accuracy, it can be added to the Service Catalog with the corresponding mapping to its implementation, and used as a service for the next pipeline design. We note that, when tuning the classifier, Alice could need to adapt the pipeline, for example changing the anonymization algorithm to obtain a better data quality. In this case, it is sufficient to replace one of the anonymization services from the catalog and re-deploy the MBDAaaS process.

3.2. Detection pipeline: classification of nosy admin activities

This pipeline works in production and, once deployed, detects (classifies) *nosy admin* attack. As in the previous case, Bob defines the high-level goals considering his business (security) needs and, possibly, additional constraints. For example, legal constraints follow the data minimization principle and require to perform pseudo-anonymization. The outcome of this step is again a declarative model representing the pipeline. This pipeline is rather simple: it sequentially composes data ingestion, pseudo-anonymization, classification, and visualization steps (see Fig. 2(b)). Bob can then define it with no (or minimal) support from Alice.

Following the MBDAaaS methodology, concrete services compatible with the declarative model specified by the user are retrieved from the service catalog, including the trained classifier and corresponding data preparation services, described in Section 3.1. From this point on, MBDAaaS methodology defines the platform-independent service composition (procedural model), transforms it into a platform-dependent workflow (deployment model), and finally deploy the platform-dependent workflow in the target execution platform. The overall process can be highly automated, with minimal human intervention, as far as the MBDAaaS compiler is correctly configured for the target big data platform.

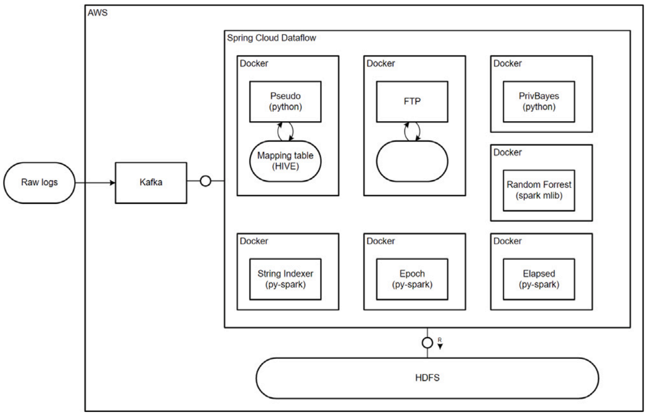


Fig. 3. Pilot Architecture.

Bob can now test and validate the application and, if some adaptation is needed, can replace or re-configure a service, quickly re-deploying a new pipeline built on the previous one. In the next section, we show how the pipeline can be easily extended to detect an additional type of attack.

3.3. Detection pipeline: detection of dormant accounts

Let us consider the scenario in which, after few months from the deployment and execution of the detection pipeline in Section 3.2, Bob realizes that a new source of attacks can affect his system through anomalous activities from *forgotten users*. In other words, dormant accounts that should not be active can be used as a driver for attacks. Sharing his expertise on the main features of the attack with the data scientist, Bob provides Alice with additional data for training a new classifier. Alice decides to include a new feature: 'elapsed time', which will contain, for each entry, how many seconds passed since the same user performed an action. The larger the value, the higher the probability that the user is dormant. She develops an additional service, called Elapsed, to pre-process the data. Using the data, she can retrain the classifier to detect the dormant attacks and, following MBDAaaS paradigm, add the service to the catalog with the corresponding implementation for the target platform. Since MBDAaaS permits model reuse, Bob can re-deploy the pipeline for the "Detection of *nosy admin* activities" (Fig. 2(b)), just replacing the classifier (including the data preparation service) (Fig. 2(c)) with small effort, and run it in production. In the next section, we present some details of the architecture supporting these pipelines and the services realizing the workflow.

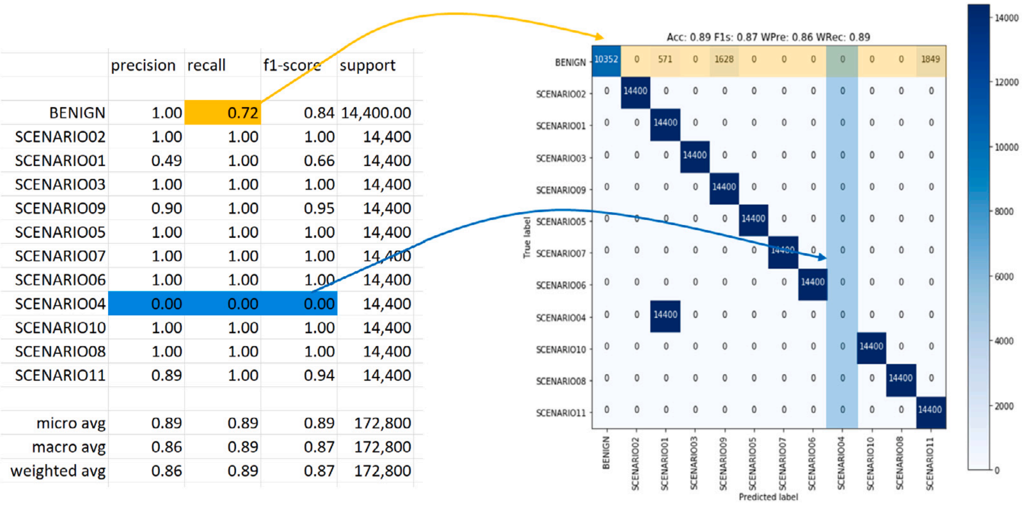
4. Architectural aspects and implementation

To demonstrate the feasibility of our approach, we implement the pipelines in Section 3 on the SAP production platform using MBDAaaS. Fig. 3 shows the architecture addressing the requirements. The pipelines are based on Spark 2 services, and are encapsulated and distributed on Docker containers. These containers are later orchestrated via Spring Cloud Dataflow using the pipeline definition generated by the MBDAaaS compiler. All services run on the AWS infrastructure with Hadoop Distributed File System.

Bootstrap pipeline (Fig. 2(a)) only requires services Pseudonymization, Ingestion, and PrivBayes; this architecture is provided as an abstraction of the different services running on the platform. Data can be ingested directly or through a Kafka pipeline. Services are deployed in their own Docker image, exposing entry points to be called by the Spring Cloud Dataflow orchestrator. HDFS storage is used to exchange data between containers.

Anonymization plays a major role in this pipeline. The overall goal is to generate the privacy-safe data for training. The process is realized in two steps: pseudo-anonymization and data publishing. In step pseudo-anonymization, personal data are replaced with random pseudonyms. The major drawback is that the correlation structure is destroyed by this process, making hard to run any analytics. Accordingly, it is only applied to the more sensitive fields in the logs: real user names and login identifiers. As shown in Fig. 3, pseudonyms mapping is maintained in the HIVE data warehouse, which permits, under certain circumstances, to revert the (pseudo)-anonymization process for further investigation of security incidents. All the other sensitive fields are anonymized using a service implementing PrivBayes, a method based on differential privacy for releasing high-dimensional data, preserving some data dependencies, that is, the Bayesian network model derived from the original data [18]. This bootstrap pipeline can be time-consuming, especially for PrivBayes process, but it is executed once, in a single batch, as its objective is creating data for training.

As mentioned in Section 3.1, before training the classifier on (anonymized) logs, Alice needs to run a data preparation step. Since the Random Forest model requires numerical vectors as input, she relies on additional services converting categorical in numerical



	precision	recall	f1-score	support
BENIGN	1.00	0.72	0.84	14,400.00
SCENARIO02	1.00	1.00	1.00	14,400
SCENARIO01	0.49	1.00	0.66	14,400
SCENARIO03	1.00	1.00	1.00	14,400
SCENARIO09	0.90	1.00	0.95	14,400
SCENARIO05	1.00	1.00	1.00	14,400
SCENARIO07	1.00	1.00	1.00	14,400
SCENARIO06	1.00	1.00	1.00	14,400
SCENARIO04	0.00	0.00	0.00	14,400
SCENARIO10	1.00	1.00	1.00	14,400
SCENARIO08	1.00	1.00	1.00	14,400
SCENARIO11	0.89	1.00	0.94	14,400
micro avg	0.89	0.89	0.89	172,800
macro avg	0.86	0.89	0.87	172,800
weighted avg	0.86	0.89	0.87	172,800

Acc: 0.89 F1s: 0.87 WPre: 0.86 WRec: 0.89

True label \ Predicted label	BENIGN	SCENARIO02	SCENARIO01	SCENARIO03	SCENARIO09	SCENARIO05	SCENARIO07	SCENARIO06	SCENARIO04	SCENARIO10	SCENARIO08	SCENARIO11
BENIGN	10352	0	571	0	1628	0	0	0	0	0	0	1849
SCENARIO02	0	14400	0	0	0	0	0	0	0	0	0	0
SCENARIO01	0	0	14400	0	0	0	0	0	0	0	0	0
SCENARIO03	0	0	0	14400	0	0	0	0	0	0	0	0
SCENARIO09	0	0	0	0	14400	0	0	0	0	0	0	0
SCENARIO05	0	0	0	0	0	14400	0	0	0	0	0	0
SCENARIO07	0	0	0	0	0	0	14400	0	0	0	0	0
SCENARIO06	0	0	0	0	0	0	0	14400	0	0	0	0
SCENARIO04	0	0	14400	0	0	0	0	0	0	0	0	0
SCENARIO10	0	0	0	0	0	0	0	0	0	14400	0	0
SCENARIO08	0	0	0	0	0	0	0	0	0	0	14400	0
SCENARIO11	0	0	0	0	0	0	0	0	0	0	0	14400

Fig. 4. Visualizing Analytics Results.

fields (String Indexer), representing dates as seconds (Epochs), and displaying (visualization service) the classifier's confusion matrix (see Fig. 4). These services are re-used in the production pipeline.

Using the output of Bootstrap pipeline and corresponding services, Alice can devise and train a Random Forest model that receives as input an entry and provides as output labels 'benign', 'nosy admin', or 'privilege escalation'.

The 'nosy admin' detection pipeline is realized using data ingestion, pseudo-anonymization, data preparation, classifier, and visualization services. Being it run by Bob without the involvement of other roles, there is no need of complex anonymization step, and basic pseudo-anonymization is sufficient. Since all these services are available, both in the catalog and in the target platform, Bob can design and deploy the security application using MBDAaaS platform, without additional support from Alice.

Similarly, to adapt the application for including detection of dormant accounts, Bob just needs to replace the previously trained model with the new one available in the platform (it was added to the catalog by Alice together with the corresponding data preparation service Elapsed) and re-deploy the pipeline.

5. Evaluation and progress with respect to the state of the art

According to software engineering research, finding near-optimal configurations for highly configurable systems, such as Software Product Lines, requires to deal with exponential space complexity [19]. This relates to the size of the combinations to be considered (the power set of the assessed features), but also to the fact that feature interactions introduce performance dependencies. In assessing performance, it is not possible to simply add the contributions provided by every single feature in a combination. Software systems need to be adapted to a variety of environments, constraints, or objectives. For this reason, software systems are designed to support different configurations. Comparing different configurations' performance is a key issue. A naive approach may consider the system as a black box and produce a set of observations for measuring the performance of the system in each specific configuration. This approach is however infeasible in every context where the number of configurations is so high that their combinations become intractable. To make performance prediction practicable two general strategies have been followed, **sampling-centric** and **model-centric**, even if several works apply a combination of the two.

Sampling-centric strategy. The goal is to generate a model from a set of samples, as restricted as possible, preserving at the same time the prediction accuracy. Random sampling is the standard operating procedure; however, a true-sampling approach should consider only valid configurations. Filtering out invalid configurations is a possible answer [20], but filtering does not avoid having to generate the entire set of configurations [21]. Encoding configurations in a feature model that uses propositional calculus can reveal inconsistent configurations, that is, incompatible combinations (conjunctions) of features in the model, avoiding to explore the entire configuration space [19]. Statistical procedures have been adopted to sample configurations to achieve high sample variance [22], if necessary in conjunction with other considerations such as bounded acquisition cost [23]. Machine Learning has been often used to generate valid samples as it permits an iterative approach where the training set contains the samples used to create the model, and the test set is used to assess this sample and enlarge it if the accuracy is low [24].

Model-centric strategy. Measuring the performance of a real system is typically time-consuming, costly, and implies risks of damaging the system itself. For this reason, model-driven approaches are sometimes preferred. For example, performance-annotated software architecture models (e.g., written in UML-2) can be transformed into stochastic models (e.g., Petri Nets) to which analytical or simulation-based optimization methods can be applied. A comparative assessment of different approaches with the trade-offs they entail is proposed in [25]. Model-centric approaches can also be used to support sampling. Feature Models can be exploited to represent the interdependencies between features in configurations restricting the sampling space to valid configurations [19]

Table 1
Comparison of security features between MBDAaaS proposal and other main platforms.

Features	Improvado	Databricks	MBDAaas	QBole	Dataminr	Sisense
Security by Design	✗	✗	✓	✗	✗	✗
Tailored for Security Data	✗	✗	✓	✗	~	✗
Security Level Customization	✗	✗	✓	✗	✗	✓
Fast Rollout	✓	~	✓	✗	✓	~
Self Service Analytics	✗	~	✓	~	✓	~
Secure Data Disclosure	✗	✓	~	✗	✓	✓

or identifying the minimum set of features required to determine a performance [26]. Clearly, not all inconsistencies are known a priori. Our approach implements a model-centric view that guarantees fast application development at the different specification levels handled by our architecture. The models we use for connecting features and performance are updated based on the results obtained by running the system and monitoring its operating results. Recent contributions in this area work on top of platform-specific configuration libraries. KeystoneML [27] introduced an approach for large-scale pipeline optimization extending Spark ML libraries. The authors focus on capturing end-to-end pipeline application characteristics that are used to automatically optimize execution at both the operator and pipeline application levels. Other work, such as [28], has focused on testing and monitoring Machine Learning models, going beyond error rate but focusing on system reliability and lowering long-term maintenance costs. A high-level data-flow abstraction for modeling complex pipelines is also proposed in [29]. The data flows proposed in this work are direct acyclic graphs that specify some aspects of a pipeline, delegating data inspections and optimization to the execution stage. In [28], the authors propose an adaptation of TensorFlow for supporting data analysis, transformation, and validation. The aim is boosting automation in the deployment of machine learning models. The main limitations of the current proposals are that they are closely tied to specific frameworks, such as Spark in [27] or TensorFlow in [28], and lack of a formal definition supporting verification procedures for BDA pipelines.

Table 1 compares our solution with the main platforms available in the market. We propose a functional comparison based on the security aspects that are deemed to be relevant from a business point of view. We took into consideration 6 main features related to security: *(i) security by design*, the possibility to formalize security at design time and automate security controls, *(ii) tailored for security data*, the native support for security data analysis including security and privacy aspects, *(iii) secure data disclosure,* the availability of ad hoc services and tools for anonymization and privacy protection, *(iv) customization of security levels,* the possibility to personalize the level of security for accessing data, including the traditional CIA triad extended with accountability and privacy, *(v) self-service analytics,* the possibility to implement and deploy a custom analytics, which permits to increase the security features, *(vi) fast roll-out,* the ability to support a fast roll-out of a data analytics campaign reducing the gap between the design phase and the deployment phase. From our analysis, we can observe that existing platforms mostly focus on personalized platform services (features 4 and 5) and fast roll-out (feature 6), while they generally fail to address architectural aspects of security (features 1, 2, and 3). This way, our approach (MBDAaaS in Table 1) addresses security expectations of a Data Analytics Pipeline-as-a-Service thanks to its model-based approach, where declarative requirements of a computation are semi-automatically compiled in a ready-to-be-executed deployment model. Also, our continuous development life cycle permits to adapt the implemented analytics to address the intrinsic dynamics of security-oriented pipelines.

6. Conclusions

The huge potential of machine learning and artificial intelligence is impaired by the difficulty of designing and deploying analytics that properly achieve the expectations of the users from both accuracy and security viewpoints. In this paper, we adopted our MBDAaaS methodology as a middleware capable to orchestrate the actors of a business process in carrying out a data analytics. Our approach has been instantiated in a security scenario typical of Smart Grids, where security experts and data scientists collaborate to instantiate an analytics aimed to detect security incidents in a privacy-preserving way through log analysis.

CRediT authorship contribution statement

Claudio A. Ardagna: Writing - review & editing, Term, Conceptualization, Methodology, Validation, Investigation, Writing - original draft. **Valerio Bellandi:** Writing - review & editing, Term, Conceptualization, Methodology, Validation, Investigation, Writing - original draft. **Ernesto Damiani:** Writing - review & editing, Term, Conceptualization, Methodology, Validation, Investigation, Writing - original draft. **Michele Bezzi:** Writing - review & editing, Software, Investigation, Writing - original draft. **Cedric Hebert:** Software, Investigation, Writing - original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by the European Union's Horizon 2020 research and innovation programme under the CONCORDIA: Cyber security cOmpeteNce fOr Research anD Innovation project, grant agreement No 830927, and by the programme "Piano sostegno alla ricerca" funded by Università degli Studi di Milano.

References

[1] Ardagna C, Ceravolo P, Damiani E. Big data analytics as-a-service: Issues and challenges. In: Proc. of PSBD 2016. 2016.
[2] Dobre C, Xhafa F. Intelligent services for Big data science. Future Gener Comput Syst 2014;37:267–81.
[3] Abellan C, Pruneri V. The future of cybersecurity is quantum. IEEE Spectr 2018;55(7):30–5.
[4] Ardagna C, Bellandi V, Bezzi M, Ceravolo P, Damiani E, Hebert C. A model-driven methodology for big data analytics-as-a-service. In: Proc. of IEEE BigData congress 2017. 2017.
[5] Gungor VC, Lu B, Hancke GP. Opportunities and challenges of wireless sensor networks in smart grid. IEEE Trans Ind Electron 2010;57(10):3557–64.
[6] Yigit M, Gungor VC, Baktir S. Cloud computing for smart grid applications. Comput Netw 2014;70:312–29.
[7] Simmhan Y, Kumbhare AG, Cao B, Prasanna V. An analysis of security and privacy issues in smart grid software architectures on clouds. In: Proc. of IEEE CLOUD 2011. 2011.
[8] Vasilakos A, Hu J. Energy Big data analytics and security: Challenges and opportunities. IEEE Trans Smart Grid 2016;7:1. http://dx.doi.org/10.1109/TSG.2016.2563461.
[9] Davenport T, Patil D. Data scientist. Harv Bus Rev 2012;90(10):70–6, cited by 106.
[10] Mauro AD, Greco M, Grimaldi M, Ritala P. Human resources for Big data professions: A systematic classification of job roles and required skill sets. Inf Process Manage 2018;54(5):807–17.
[11] Miller S. Collaborative approaches needed to close the big data skills gap. J Organ Des 2014;3(1):26–30, cited by 23.
[12] Ardagna C, Bellandi V, Bezzi M, Ceravolo P, Damiani E, Hebert C. Model-based Big data analytics-as-a-service: Take Big data to the next level. IEEE Trans Serv Comput 2018.
[13] Ardagna C, Bellandi V, Ceravolo P, Damiani E, Martino BD, D'Angelo S, Esposito A. A fast and incremental development life cycle for data analytics-as-a-service. In: Proc. of IEEE BigData congress 2018. 2018.
[14] Yamin MM, Katt B, Sattar K, Ahmad MB. Implementation of insider threat detection system using honeypot based sensors and threat analytics. In: Arai K, Bhatia R, editors. Advances in information and communication. Springer International Publishing; 2020, p. 801–29.
[15] Gunduz MZ, Das R. Cyber-security on smart grid: Threats and potential solutions. Comput Netw 2020;169:107094.
[16] Hinatsu S, Shimizu K, Ueda T, Boyer B, Mentré D. Automatic vulnerability identification and security installation with type checking for source code. In: Barolli L, Nishino H, Enokido T, Takizawa M, editors. Advances in networked-based information systems. Springer International Publishing; 2020, p. 292–304.
[17] Leida M, Ceravolo P, Damiani E, Asal R, Colombo M. Dynamic access control to semantics-aware streamed process logs. J Data Semant 2019;8(3):203–18.
[18] Wood A, Shpilrain V, Najarian K, Kahrobaei D. Private naive bayes classification of personal biomedical data: Application in cancer data analysis. Comput Biol Med 2019;105:144–50.
[19] Oh J, Batory D, Myers M, Siegmund N. Finding near-optimal configurations in product lines by random sampling. In: Proc. of ESEC/FSE 2017. 2017.
[20] Sayyad AS, Menzies T, Ammar H. On the value of user preferences in search-based software engineering: a case study in software product lines. In: Proc. of IEEE/ACM ICSE 2013. 2013.
[21] Henard C, Papadakis M, Harman M, Le Traon Y. Combining multi-objective search and constraint solving for configuring large software product lines. In: Proc. of IEEE/ACM ICSE 2015. 2015.
[22] Guo J, Czarnecki K, Apel S, Siegmund N, Wasowski A. Variability-aware performance prediction: A statistical learning approach. In: Proc. of IEEE/ACM ASE 2013. 2013.
[23] Sarkar A, Guo J, Siegmund N, Apel S, Czarnecki K. Cost-efficient sampling for performance prediction of configurable systems (t). In: Proc. of IEEE/ACM ASE 2015. 2015.
[24] Jamshidi P, Casale G. An uncertainty-aware approach to optimal configuration of stream processing systems. In: Proc. of IEEE MASCOTS 2016. 2016.
[25] Brosig F, Meier P, Becker S, Koziolek A, Koziolek H, Kounev S. Quantitative evaluation of model-driven performance analysis and simulation of component-based architectures. IEEE Trans Softw Eng 2015;41(2):157–75.
[26] Schröter R, Krieter S, Thüm T, Benduhn F, Saake G. Feature-model interfaces: the highway to compositional analyses of highly-configurable systems. In: Proc. of IEEE/ACM ICSE 2016. 2016.
[27] Sparks ER, Venkataraman S, Kaftan T, Franklin MJ, Recht B. KeystoneML: Optimizing pipelines for large-scale advanced analytics. In: Proc. of IEEE ICDE 2017. 2017.
[28] Baylor D, Breck E, Cheng H-T, Fiedel N, Foo CY, Haque Z, et al. TFX: A TensorFlow-based production-scale machine learning platform. In: Proc. of ACM SIGKDD 2017. 2017.
[29] Böse J-H, Flunkert V, Gasthaus J, Januschowski T, Lange D, Salinas D, et al. Probabilistic demand forecasting at scale. Proc VLDB Endow 2017;10(12):1694–705.

Claudio A. Ardagna is a professor at the Department of Computer Science, Università degli Studi di Milano. His research interests are in the areas of big data, artificial intelligence, and cloud/edge security and assurance. He is the recipient of the ERCIM STM WG 2009 Award for the Best Ph.D. Thesis on Security and Trust Management. He has co-authored the Springer book "Open Source Systems Security Certification".

Valerio Bellandi received the Ph.D. in computer science at the University of Milan. He is currently assistant professor at the Computer Science Department. His research interests are focused on a Smart Data Driven Systems. He is focused both from algorithmic and architectural point of view. He has been the guest editor of over 20 special and he is involved in several research projects.

Ernesto Damiani is Full Professor at Università degli Studi di Milano, Senior Director of Robotics and Intelligent Systems Institute and Director of Center for Cyber Physical Systems (C2PS) within the Khalifa University, leader of the Big Data area at Etisalat British Telecom Innovation Center, and President of the Consortium of Italian Computer Science Universities (CINI).

Michele Bezzi is Research Manager at SAP Security Research. He holds a Ph.D. in Physics at University of Bologna. He has 15+ years' experience in industrial research in SONY, Accenture and SAP. He has been supervising several European projects, and he published 50+ scientific papers in : security, privacy, machine learning, neural networks, evolutionary models, complex systems.

Cédric Hebert is a Senior Researcher at SAP Security Research, where he leads Active Defense research. As a certified infosec expert, his work ranges from secure design to offensive security. Some of his current and past projects include the creation of SAP's Enterprise Threat Detection solution, blockchain security and anonymization technologies for Big Data analytics.