International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Prediction Model for Occupational Incidents in Chemical and Gas Industries

Ganapathy Subramaniam Balasubramanian, Ramaprabha Thangamani

Abstract*: Understanding occupational incidents is one of the important measures in workplace safety strategy. Analyzing the trends of the occupational incident data helps to identify the potential pain points and helps to reduce the loss. Optimizing the Machine Learning algorithms is a relatively new trend to fit the prediction model and algorithms in the right place to support human beneficial factors. The aim of this research is to build a prediction model to identify the occupational incidents in chemical and gas industries. This paper describes the architecture and approach of building and implementing the prediction model to predict the cause of the incident which can be used as a key index for achieving industrial safety in specific to chemical and gas industries. The implementation of the scoring algorithm coupled with prediction model should bring unbiased data to obtain logical conclusion. The prediction model has been trained against FACTS (Failure and Accidents Technical information system) is an incidents database which have 25,700 chemical industrial incidents with accident descriptions for the years span from 2004 to 2014. Inspection data and sensor logs should be fed on top of the trained dataset to verify and validate the implementation. The outcome of the implementation provides insight towards the understanding of the patterns, classifications, and also contributes to an enhanced understanding of quantitative and qualitative analytics. Cutting edge cloud-based technology opens up the gate to process the continuous in-streaming data, process it and output the desired result in real-time. The primary technology stack used in this architecture is Apache Kafka, Apache Spark Streaming, KSQL, Data frames, and AWS Lambda functions. Lambda functions are used to implement the scoring algorithm and prediction algorithm to write out the results back to AWS S3 buckets. Proof of concept implementation of the prediction model helps the industries to see through the incidents and will layout the base platform for the various safety-related implementations which always benefits the workplace's reputation, growth, and have less attrition in human resources.

Keywords*: Occupational incidents, Prediction Model, Machine Learning, Real-time processing.

I. INTRODUCTION

R esearch and Development department of organizations and industries might have challenges when they try to implement a solution. There might be a lot of unknowns when providing the input to relate to the concept along with computational constraints. Implementation of new algorithms and new models have a similar step-up process to verify and validate the real-time scenarios. The Scoring algorithm [15] and SafeOne[1] Prediction Model for Support Vector Machines (SVM) that is developed for prediction of occupational incidents, definitely need an architecture to get through the acceptable implementation. In-flow data should constantly monitor to compute the exact score and provide the expected output. Developing the proof of concept (POC) will help the organization to see-through the potential outcome of the solution and also helps to identify the gaps in it [23]. It will also provide the stakeholders to internally evaluate the promising solution which helps to reduce the unnecessary risk. Design expectations and potential timeline can also be determined before the full-scale implementation. Applying a defined algorithm is not an easy task. The workflow should be determined and a state of maintenance should be established. As a part of POC, it is necessary to build an interface to visualize the best possible results [7][12].

II. PROCESS MODEL

An evolutionary prototyping process model, as shown in Fig. 1, is being used to build this proof of concept (POC) which constantly helps to refine the solution [10]. Since the implementation of the Scoring model and SafeOne prediction model is complex and preliminary outcome is unclear and sporadic. The prototyping model helps to evaluate the accuracy of the results, factors to be modified, missing factors and remove the outliers. This helps to improve the overall solution while they are being built. Algorithms are never done, but it always maturing as the application of the factors and in-flow data changes [2][12].

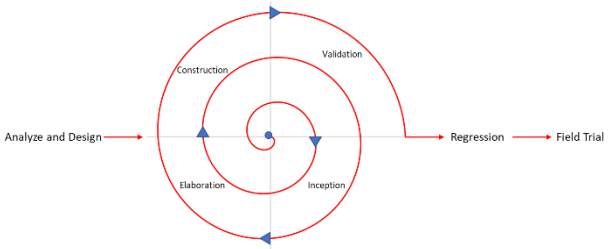

Fig. 1. Evolutionary prototyping process model (Spiral model)

[1] SVM Model is being developed as a part of Research & Development

Manuscript published on November 30, 2019.
* Correspondence Author
Ganapathy Subramaniam Balasubramanian*, Research Scholar, PG and Research Department of Computer Science and Applications, Vivekanandha College of Arts and Science, TN, India.
Dr. T. Ramaprabha, PG and Research Department of Computer Sciences and Applications, Vivekanandha College of Arts and Science, Tiruchengode, TN, India.

A. Inception

Requirements and Use cases have been well defined to achieve the outcome of the algorithms. The primary goal is to predict the particular zone where it may have the potential to expose to the occupational incident by classifying the zone between Danger, Caution and Good. Continuous streaming of inspection data and sensor data flows through the pipeline.

In this case, producer and consumer part of the programs uses the JSON format to send the data into the workflow. Algorithms are defined through Spark library which can evaluate the data in the system. The output should be in the form of graphical representation.

B. Elaboration

Overall analysis and design, detailed iteration plan, technical specification, and functional specification documents are the key deliverables in this phase. Work items to format the input inspection data and sensor data, parsing the data into the structured model to store it in the database, retrieval methodology from the database to process the score and graphical structure and model are defined as a granular tasks where it can be independently developed, validated and deployed as a complete component.

C. Construction

Development activities to put the design ideas of pipeline processing, evaluating the data from the pipeline, calculating the score using the Scoring algorithm and final computation to predict the zone and plot the graph as a complete visual workable solution have been done in the phase.

D. Validation

Verification and validation of the score and the zone and type of the occupational incident prediction have been continuously monitored for the varying outcome. Validation makes the defect-free evaluation to present the clear output value to proceed for the next step of operations. Continuous testing and Continuous deployment have been configured using Selenium and Jenkins respectively.

III. ARCHITECTURE

The architecture of the POC, as shown in Fig 2., detailed out the components involved for the better outcome. The integration between these components is aligned in such a way to establish a scalable solution for the future data load. Starting from data stream through getting the outcome of the prediction model, cloud infrastructure helps to deliver a reliable solution.

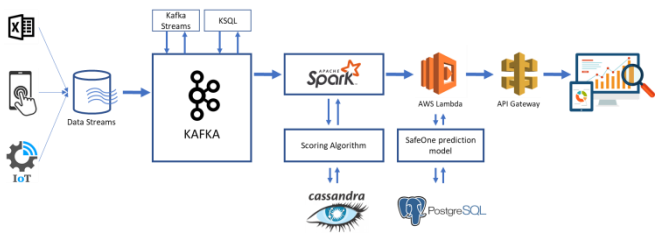

Fig 2. High-level architecture diagram

Data Streaming is a method of posting a continuous stream of data that can be processed through the algorithms to obtain structural data. Multiple sources can send the data simultaneously to meet the requirements of real-time data analytics. Inspection data, sensor data from the gas detection instrument and historical incidents should be streamed through Kafka. The continuous stream of data is put in a bucket called topic. Topics in Kafka can be subscribed by the consumer program to stream for processing [11]. These topics are partitioned based on the size and volume of speed and scalability.Data are sent by various data sources to topics and subscribed consumer application takes care of relaying it. Each partition is assigned to a Kafka Broker for parallel processing [16]. Messages are typically key-value pairs so as to construct the structural data.

The stream is divided into RDDs (Resilient Distributed Datasets) which is a fundamental data structure of Spark. RDDs are divided into partitions which consist of tuples. The worker node takes care of processing the data in the Spark. Kafka-Spark connector allows mapping partitions between RDD and Kafka topic [17].

Processing of the data takes place through Spark Jobs. Spark Jobs is the small set of programs that cleans up, manipulates and applies the specific algorithm to the data streamed and stored into the datalake. Datalake is a collection of data frames stored in the storage bucket [16]. Spark Jobs written using Scala language in Notebook executes the Scoring algorithm to refine and restructure the data which should be used as an input for the SafeOne prediction model. Lambda functions serve the purpose of executing the logic using the structured data to provide the expected outcome. The approach of incremental algorithms can be used to manipulate the history data and real-time data. Heatmap representation of the data can be generated from algorithm to visualize the results. The data dashboard displays the required heatmap and also keeps the data live through the push notifications.

IV. IMPLEMENTATION

Each participating industry pushes the inspection data in the tail of a central messaging queue to be processed. Such a data processing queue allows to decouple the emitting applications from the receiving ones, but also to take over the peaks of inspection data by smoothing the load for processing applications. Then, the data is stored in an immutable database. Each input such as an inspection data is composed of meta-information such as the inspection type, description, completed on date, inspection status, case number, etc., The sample data below illustrates an inspection data in JSON (JavaScript Object Notation):


```
[
 {
   "the_geom": "0101000020E610000017E12C0695C752C0ECE9AF581C054440",
   "lng": -75.1184707105898,
   "objectid": 34,
   "addresskey": 1430,
   "opa_account_num": 612496600,
   "the_geom_webmercator": "0101000020110F0000D3AF277A29E65FC1798F148990955241",
   "aptype": "CD ENFORCE",
   "apdesc": "CODE ENFORCEMENT UNITS",
   "apinspkey": 1492636,
   "inspectiontype": "HCEU INSP",
   "inspectiondescription": "CE-HOUSING CODE ENFORCE INSP",
   "inspectioncompleted": "2011-07-08 17:08:00",
   "inspectionstatus": "Failed",
   "geocode_x": 2705207.52935076,
   "geocode_y": 268379.56271823,
```

Data is directly processed by a calculation engine for the appropriate data to be extracted from the Scoring algorithm to clean up and fill in the data with the cause of the incident for the unknown ones. The result of this algorithm along the way updates the view, offering a real-time prediction of the actual outcome occupational incident.

On a periodic basis, a batch will be run to calibrate the data to correct the deviations. Excerpt of the code to read the Kafka stream and creating the direct stream of Spark is shown below:

```
// The stream is splitted into RDDs, each part representing a 2 seconds interval
val streamContext = new StreamingContext(sparkConf, Seconds(2))
// We subscribe to the "receipts" topic
val topicsSet = Set("inspectionData")
// Kafka brokers list
val kafkaParams = Map("metadata.broker.list" -> "kafka1:9092,kafka2:9092")
// Spark stream from Kafka stream
val lines = KafkaUtils.createDirectStream[String, String,
    StringDecoder, StringDecoder](ssc, kafkaParams, topicsSet)
// We are now able to deploy the scoring algorithm
```

SafeOne prediction model is being continuously trained using data streams through Spark is shown as an output of this proof of concept. Data is usually an hourly snapshot from the FACTS database and from the industrial inspection data feed. For each hour, the factors considered for the classifications and models are given for the manipulation of data. Based on the data, SafeOne prediction model has been trained to predict the potential value of the occupational incidents in the industry. Once model is trained, test set of data has been applied for verification and validation of the model to write the prediction value. This POC stores the trained data into S3 bucket. Model creation and prediction results are stored into PostgreSQL database in a batch mode.

AWS Lambda serves as a good candidate to relay the results of the prediction model from the database without any hindrances around scalability, portability, and security. As the same, Lambda functions provide the prediction results to other appropriate data storage systems for further proceedings [17]. This approach helps not only to regression models but also used to build an analytical or a classification system. However, it should be noted that there should be some latency between the real-time model update and visualization of the prediction results from the database.

V. RESULT AND DISCUSSION

The calculated data should be visualized in order to understand the results of prediction. Dashboard UI page created using HTML and JS (JavaScript) libraries helps to showcase the data as a visual element. JS code receives the data through API Gateway and parses the response data in JSON format. ChartJS renders the graph in real-time when the processed data appears in the cloud database. SafeOne model predicts the "Unsafe" percentage with reference to all the factors considered during the machine learning process. The radar style graph is used to plot the score of each inspection type. Value 0 being the most unsafe zone and value 100 being the safest zone on the radar.

The sample structure of the processed inspection data with its score value obtained from one of the organizations is shown below for the reference. Though the inspection data are collected on the daily or frequent shift basis, bi-weekly sample data is shown here for better representation. Inspection types and timelines are given as the row and column headings respectively.

	1/6/2010	1/20/2010	2/3/2010	2/17/2010
Employee Observation / PPE	82.1	83.1	85.1	77.9
Housing Keeping	31.9	17.2	75.3	30.5
Railings / Covers / Cables	75.0	29.0	31.3	20.3
Fire protection	82.8	88.3	78.5	89.0
First aid / Sanitation	28.2	77.7	66.0	53.7
Tools	32.3	13.3	32.1	31.5
Scaffoldings	26.3	75.0	30.5	28.4
Ladders and Stairs	33.5	26.3	29.7	72.7
Electrical	74.0	34.8	80.8	73.5
Lifts	30.0	16.0	28.2	14.7
Programs / information	34.5	71.8	77.5	33.9
Site / Public & Environmental protection	35.0	67.8	69.5	68.8

Fig 3. Sample processed inspection data with its score

The outcome of the Prediction model should be validated against the known results to verify against the obtained result in the past. The machine learning model has been trained in such a way to produce the result from the learning so far. The dashboard has been set up to refresh by itself for every five seconds. Following dashboard displays the radar graph along with Unsafe percentage:

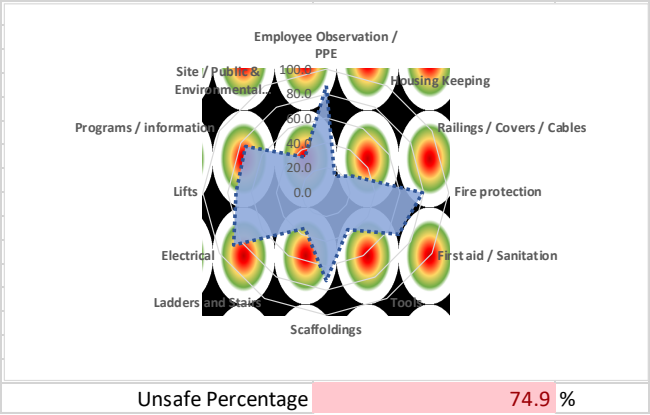

Fig 4. Sample dashboard with higher Unsafe percentage

Another example prediction which shows lower Unsafe percentage is shown below:

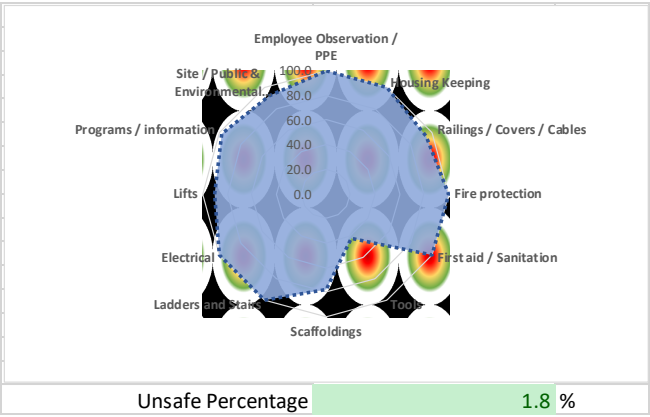

Fig 5. Sample dashboard with lower Unsafe percentage

By having regular health checks of the model and the continuous performance monitoring improves the quality of the result. Having the resulting dashboard to be responsive, it can be viewed on different devices and resolutions. Training data was split into 70% which is 754,660 rows of training set data and 229,000 rows for the test set.

An overall error has also been calculated from the results as a part of the prediction model calculation. Though the overall prediction did not show the best performance in predicting occupational incidents outcome, the model has a considerable accuracy which can definitely be enhanced in the near future. These results are significant and provide a positive look forward solutions to practice safety in the organizations to prevent potential incidents. This proof of concept provides the basic idea of visualizing the results in real-time by setting the base platform to smoothly walk through the workflow from raw data to prediction results.

VI. CONCLUSION

Prototyping the prediction model in real-time technology serves the clarity of the requirements and improves the insight by resulting in better implementation. The presence of the proof of concept prevents many miscalculations and deviations in the training data. The spiral process model structures the roadmap of the implementation by defining the tasks in Inception, Elaboration, Construction and Validation phases. Architecture has been defined in such way to proceed with the seamless workflow from the input stream to execution of the prediction model to obtain the final result. Implementing the real-time data streams through Kafka and Spark helps to evaluate the performance of the incoming data and improvises the process flow in prediction model for continuous validation. A score set of the inspection data has been obtained through the Scoring algorithm which serves as an input for the prediction model. Result visualization has been depicted using the Radar graph along with the unsafe percentage. The overall design of this proof of concept provides the fully functional working model of the proposed architecture in place which allows scaling the solution in real-time for the bigger cause.

Once the model starts working with the data to produce a considerable result, factors and parameters can be improvised in the model to try out the better classification and prediction. Prototyping of prediction model helps for better decision making on visualization, detailed reports, in turn, helps business improves their decisions on the implementation. This really helps to evaluate the accuracy of the model and predicting strategies.

REFERENCES

1. Aloysius, G., & Binu, D. (2013). An approach to products placement in supermarkets using PrefixSpan algorithm. Journal of King Saud University - Computer and Information Sciences, 25(1), 77–87. https://doi.org/10.1016/j.jksuci.2012.07.001
2. Baranyi, J., & Buss da Silva, N. (2017). The use of predictive models to optimize risk of decisions. International Journal of Food Microbiology, 240, 19–23. https://doi.org/10.1016/j.ijfoodmicro.2016.10.016
3. Bertke, S. J., Meyers, A. R., Wurzelbacher, S. J., Measure, A., Lampl, M. P., & Robins, D. (2016). Comparison of methods for auto-coding causation of injury narratives. Accident Analysis & Prevention, 88, 117–123. https://doi.org/10.1016/J.AAP.2015.12.006
4. Bureau of Indian Standards. (2007). Occupational Health and Safety Management Systems. 2000, 1–28. https://doi.org/10.1002/0471435139.hyg049.pub2
5. Davoudi Kakhki, F., Freeman, S. A., & Mosher, G. A. (2019). Evaluating machine learning performance in predicting injury severity in agribusiness industries. Safety Science, 117, 257–262. https://doi.org/10.1016/j.ssci.2019.04.026
6. Gueniche, T., Fournier-viger, P., Raman, R., & Tseng, V. S. (2015). CPT + : A Compact Model for Accurate Sequence Prediction.
7. Helvert, M. van. (2014). Data visualizations in popular Dutch media. Retrieved from Masters of Media website: http://mastersofmedia.hum.uva.nl/2014/04/17/data-visualizations-in-popular-dutch-media/
8. JANICAK, C. A. (1996). Predicting accidents at work. 115–121.
9. Kaur, P., Singh, M., & Josan, G. S. (2015). Classification and Prediction Based Data Mining Algorithms to Predict Slow Learners in Education Sector. Procedia Computer Science, 57, 500–508. https://doi.org/10.1016/j.procs.2015.07.372
10. Khan, F. I., Abbasi, S. ., Mpp, C. De, & European Union. (2006). Techniques and methodologies for risk analysis in chemical process industries. In Journal of Loss Prevention in the Process Industries (Vol. 11). https://doi.org/10.2790/73321
11. Kroß, J., Brunnert, A., Prehofer, C., Runkler, T. A., & Krcmar, H. (2015). Stream Processing on Demand for Lambda Architectures. In M. Beltrán, W. Knottenbelt, & J. Bradley (Eds.), Computer Performance Engineering (pp. 243–257). Cham: Springer International Publishing.
12. Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. Journal Of Statistical Software, 28(5), 1–26. https://doi.org/10.1053/j.sodo.2009.03.002
13. M, Suriyanarayanan., & G, Swaminathan. (2000). INDIAN CHEMICAL INDUSTRY ACCIDENT DATABASE – AN EFFORT BY CISRA M Surianarayanan * and G Swaminathan Cell for Industrial Safety and Risk Analysis Chemical Engineering Department Chennai 600020.
14. Mooney, C. H., & Roddick, J. F. (2013). Sequential pattern mining -- approaches and algorithms. ACM Computing Surveys, 45(JUNE), 1–39. https://doi.org/10.1145/2431211.2431218
15. Mr. Ganapathy Subramaniam B, Dr. T. Ramaprabha., (2019). New Scoring Algorithm for predicting the cause of incident occurrences in chemical industries. International Journal of Emerging Technologies and Innovative Research JETIR, (Vol. 6-Issue 5 (May-2019)). Retrieved from http://www.jetir.org/view?paper=JETIR1905D50
16. Qadah, E., Mock, M., Alevizos, E., & Fuchs, G. (2018). A distributed online learning approach for pattern prediction over movement event streams with apache flink. CEUR Workshop Proceedings, 2083, 109–116.
17. Qadah, E., Mock, M., Alevizos, E., & Fuchs, G. (2018). Lambda Architecture for Batch and Stream Processing. CEUR Workshop Proceedings, 2083(October), 109–116. Retrieved from https://d1.awsstatic.com/whitepapers/lambda-architecure-on-for-batch-aws.pdf
18. Reurings, M., & Janssen, T. (2006). Accident prediction models for urban and rural carriageways. 81.
19. Safety, C. S., & Hazards, C. S. (2017). FACT SHEET » Confined Spaces.
20. Sarkar, S., Pateshwari, V., & Maiti, J. (2017). Predictive model for incident occurrences in steel plant in India. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), (July), 1–5. https://doi.org/10.1109/ICCCNT.2017.8204077
21. T. Akomolafe, D., & Olutayo, A. (2013). Using Data Mining Technique to Predict Cause of Accident and Accident Prone Locations on Highways. American Journal of Database Theory and Application, 1(3), 26–38. https://doi.org/10.5923/j.database.20120103.01
22. Verma, S., & Chaudhari, S. (2017). Safety of Workers in Indian Mines: Study, Analysis, and Prediction. Safety and Health at Work, 8(3), 267–275. https://doi.org/10.1016/j.shaw.2017.01.001
23. Yael Gavish. (2017). Developing a Machine Learning Model From Start to Finish. Retrieved March 3, 2019, from Medium.com website: https://medium.com/@yaelg/product-manager-guide-part-3-developing-a-machine-learning-model-from-start-to-finish-c3e12fd835e4

AUTHORS PROFILE

Ganapathy Subramaniam B., from Chennai and born in 1979, research scholar, from Vivekanandha College of Arts and Science affiliated to Periyar University, in the field of Computer Science since Feb 2017. Data mining is his major field of research. He has obtained a Masters of Philosophy in Computer Science from Alagappa University in the year 2006 and completed his Master of Science in Information Technology in 2002. He has earned a bachelor's degree in Commerce from

Annamalai University in 2000 and has also obtained a Diploma in Computer Technology from All India Council of Technical Education in 1997. Possess good knowledge in cutting edge technology stack and his field of study is data mining in gas detection domain.

Dr. T. Ramaprabha, M.Sc., M.Phil., Ph.D., works as a Professor of PG and Research Department of Computer Science and Applications in Vivekanandha College of Arts & Sciences for Women (Autonomous), Tiruchengode, Tamil Nadu. She has completed her Ph.D., in the year 2013 at Mother Theresa Women's University, Kodaikanal. She has more than 22 years of teaching and research experience and she has published many papers in conferences and journals. Her research of interest is stereo pair image compression in image processing.