Defence Science Journal, Vol. 70, No. 6, November 2020, pp. 619-625, DOI : 10.14429/dsj.70.15545

Analysis and Prediction of Alerts in Perimeter Intrusion Detection System

Rizul Aggarwal*, Anjali Goswami, Jitender Kumar, and G.A. Chullai

Bharat Electronics Limited, Central Research Laboratory, Ghaziabad, India
**E-mail: rizulaggarwal@bel.co.in*

ABSTRACT

Perimeter surveillance systems play an important role in the safety and security of the armed forces. These systems tend to generate alerts in advent of anomalous situations, which require human intervention. The challenge is the generation of false alerts or alert flooding which makes these systems inefficient. In this paper, we focus on short-term as well as long-term prediction of alerts in the perimeter intrusion detection system. We have explored the dependent and independent aspects of the alert data generated over a period of time. Short-term prediction is realized by exploiting the independent aspect of data by narrowing it down to a time-series problem. Time-series analysis is performed by extracting the statistical information from the historical alert data. A dual-stage approach is employed for analyzing the time-series data and support vector regression is used as the regression technique. It is helpful to predict the number of alerts for the nth hour. Additionally, to understand the dependent aspect, we have investigated that the deployment environment has an impact on the alerts generated. Long-term predictions are made by extracting the features based on the deployment environment and training the dataset using different regression models. Also, we have compared the predicted and expected alerts to recognize anomalous behaviour. This will help in realizing the situations of alert flooding over the potential threat.

Keywords: Time-series; Data preprocessing; Perimeter intrusion detection system; Feature extraction

1. INTRODUCTION

Security is one of the key concerns in the current world. Advanced surveillance systems[1] are used widely for the detection of suspicious activities in real-time. Perimeter intrusion detection systems[2] play a significant role in ensuring the safety and detection of an intruder, attempting to breach a perimeter. These systems mainly rely on the sensory data for the detection of attacks or suspicious situations. Such systems focus on the alerts generated by the sensors independently; despite of having a logical connection between them. The systems are likely to cause false alerts under various scenarios. Generally, most of the generated alerts are not useful for the operator, which makes the alert systems inefficient. In certain scenarios, when there are actual intrusions not only the actual alerts will get mixed with the false alerts but also a large number of alerts become unmanageable. As a result, it becomes difficult for human users to take quick action or response in advent to the alerts generated. Also, responding to false alarms creates a burden on the operators. The major factor in understanding the intrusion systems relies on the deployment environment. By making a proper understanding of the deployment environment it is possible to:

(i) Analyze the alerts
(ii) Understand the nature of alerts generated.

The alert data obtained can be analyzed to draw the inferences about the nature of alerts, to understand what causes the false alerts in the deployed system. The case study considered for the analysis includes the perimeter intrusion detection system consisting of the sensors at the military base station. As the sensors are employed under the heterogeneous environmental conditions it results in several false alerts. In this paper, we have employed two different techniques, firstly to understand the dependency of weather conditions (as the deployed environment is open space) on the generation of alerts and other to analyse the data based on the time at which alerts were generated considering it as a time-series forecasting[3] problem. The high-level architecture of the system is shown in Fig. 1.

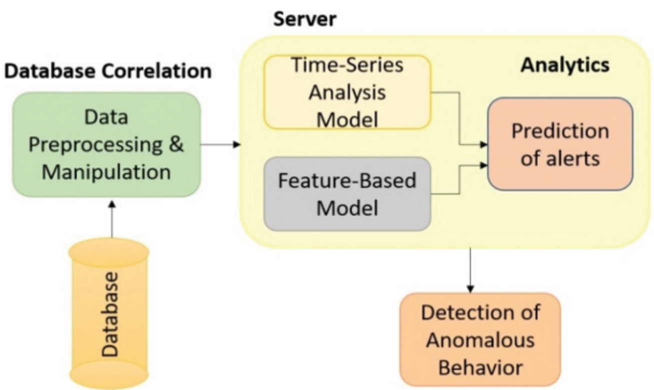

Server
Database Correlation
Data Preprocessing & Manipulation
Time-Series Analysis Model
Analytics
Prediction of alerts
Feature-Based Model
Database
Detection of Anomalous Behavior

Figure 1. High-level architecture of system.

Received : 13 March 2020, Revised : 27 June 2020
Accepted : 27 July 2020, Online published : 12 October 2020

Thus, the approach employed is useful in the short and long- term prediction of alerts in the perimeter intrusion detection system. The short-term predictions are made for next 24 hours while the long-term prediction is capable of realizing the scenario of alerts, few days ahead (depending on the availability of weather forecast data). To realize long-term prediction of alerts we have characterized the dependent aspect of the data by identification and extraction of the environmental features (temperature, season). A feature vector is created for training the data using various classifiers (Random Forest, Decision Trees). To create a feature vector, the weather data is scraped from the internet corresponding to the historical data available. Cross-validation was applied to compute the accuracy of the classifier under test. On the other hand, to understand the data on basis of time at which the alerts are generated, a hybrid approach is proposed for short-term alert predictions.

These systems are essential to build sound models to reduce the false alarms in the already deployed sensor environment at the military base station The models are built and are validated on the dataset making the following contributions:

- Comprehensive and detailed analysis of the deployment environment affecting the number of alerts generated in order to make long-term predictions.
- Proposed a dual-stage fusion approach in order to make short-term predictions for the alerts.

2. RELATED WORK

Various techniques are available to solve the problems pertaining to the alert prediction in sensor-based security systems. Most of the methodologies are patented. Trainor[4], *et al.* collects the sensor data from multiple devices and analyses the alerts based on unsupervised learning algorithms to determine a false alert situation. Kapuschat[5], *et al.* analyzes the recorded signal activity of the sensors to determine the threshold values. It generates the reports based on the comparison when signal activity from the sensors exceeds the determined threshold value. Trundle[6], *et al.* handles the alert events based on the alarm probability. The probability is calculated based on the historical data aggregated from the sensory data. Adonailo[7], *et al.* takes into consideration the security system which uses the weather data for reducing the false alert. It studies the effect of weather conditions on different type of sensory data. In this paper, a dual stage approach is proposed for prediction of alerts using Support Vector Regression (SVR). SVR is capable of providing outstanding non-linear performance, it has been applied effectively in many fields, including finance[8], daily traffic peak flow management[9–11] , electrical load forecasting[12–14] and rainfall forecasting problems[15,16].

The literature available, in our knowledge is mostly problem-centric w.r.t. alert flooding and false alerts. We have not come across any literature related to the problem catered in this paper and the techniques applied have been previously used for various domains but are novel for this problem area. We have employed a hybrid technique focusing on different aspects of the alert data generated at the military base stations.

3. DATA PREPROCESSING

The real-time data is logged at the server when an alert is generated by the sensor. The data logged consists of: *(a) Timestamp:* the reported time of the alert *(b) Sensor Type:* the type of sensor from which alert is reported *(c) Position:* the geospatial coordinates of the reported position. To predict the alerts in the nth hour, the data needs to convert into the appropriate format for further analysis. The long-term prediction can generate the overall scenario for the future days. The data was preprocessed in accordance with the input required for different models employed in this paper.

3.1 Conversion to Time-series data

The key idea behind data preprocessing is quite simple: the number of alerts varies with time as shown in Fig. 2. A higher number of alerts are observed during the early hours of the day (due to more activity of people, morning-walk). As the problem is built around prediction of alerts in the nth hour, the historical data is segregated for every hour. When the numbers of alerts are obtained for each hour, a window of n hours is considered for fetching the parameters for feature set to build training models. This can be comprehended in a way, that if we want to predict the alerts for any value of n =1, 2, 3...24, a window of n is defined to fetch the statistical information based on which a model is trained for the prediction of alerts. Figure 3 shows the processed data.

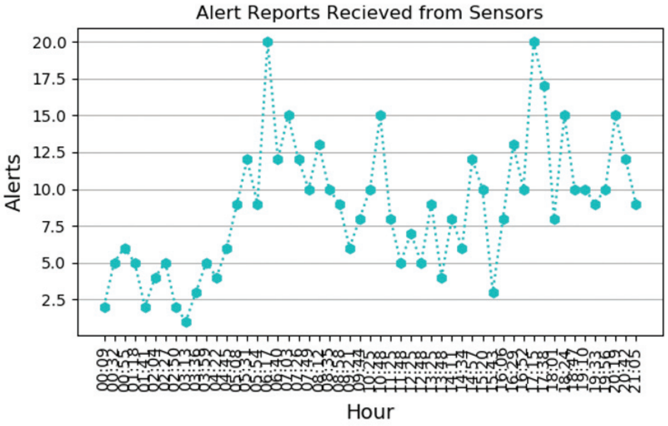


Figure 2. Alert reports logged at server.

3.2 Data Preprocessing for Long-term Prediction

Long-term prediction is done completed by understanding the dependent aspect of data i.e. the deployment environment. The data is divided into slots for different parts of the day. The alerts are segregated to obtain the alert count as: (a) Early Morning: 5am-8am (b) Late Morning: 8am- 12noon (c) Afternoon: 12noon-16pm (d) Evening: 16pm-20pm (e) Night: 20pm-5am The segregation will help in obtaining the long-term prediction of the alerts. The numbers of alerts are calculated for these '5' defined windows and correspondingly the environmental features (shown in Fig. 4(e)) are fetched. The dependency of the weather conditions is depicted in Fig. 4. It can be noticed that several features have different impact on the number of alerts obtained. As an instance, as the temperature during the winter increases, the numbers of alerts obtained are

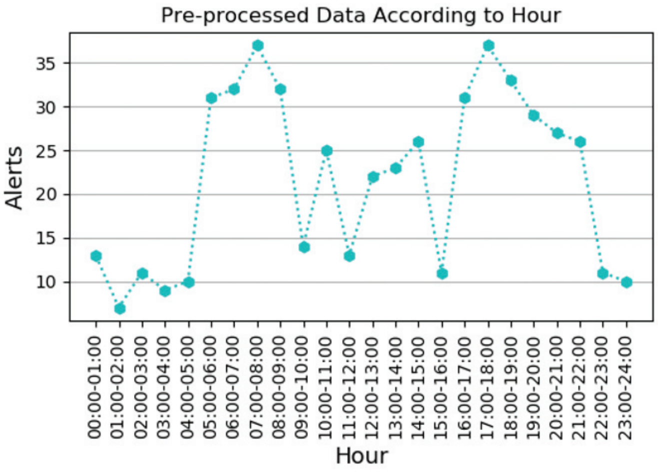


Figure 3. Processed data for time-series analysis.

high. Also, the numbers of alerts are greater during the late hours of the day (high activity of people, evening walk).

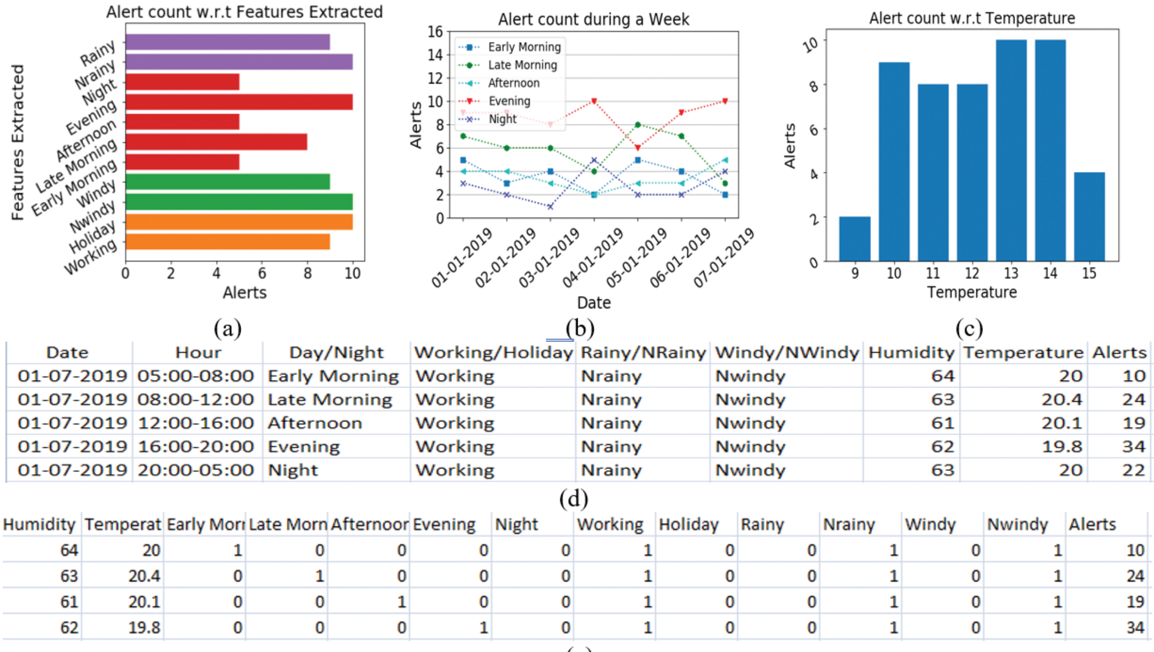
4. PREDICTION OF INTRUSION ALERTS

We first investigate the nature of data and the deployment environment. To understand this, we collected the data at the military base station consisting of different sensors to find the intrusion activities. The sensors included the underground wiring for perimeter intrusion detection, electric fencing which generates alerts when the object strikes the fence, Radar to identify the targets and a camera for image processing to understand the targets identified by Radar. These sensors generate the alerts independently, which in various conditions leads to a higher number of unmanageable alerts. Such alerts can cause *alert flooding*. The major challenge in this work is:(a) To handle and comprehend the a large number of alerts (b) To understand the aspects causing a large number of alerts. The dependent and independent aspects of the alert data were analyzed to detect anomalous activities in order to make short as well as long-term alert predictions.

4.1 Feature-based Training of Data

To model long-term prediction of alerts, the dependent aspect of data is exploited for which the features from data are extracted. The features must be dependent on the deployment environment, so the major focus is on the weather conditions. The weather data is scraped online for the days available at the deployed location. The scraped data[17] includes the features tabulated in Fig. 4(d). It is observed that some features are directly scraped like temperature, rainfall, and wind conditions while some features are derived from the scraped features. The data is scraped on the basis of the partitioning for the day (discussed in Section 3.2). It was observed that the data was available at difference of particular hours. So, the data scraped is interpolated in accordance with the previous hour to generate a feature dataset for training. The instance of the database can be observed in Fig. 4(d). The features are trained using Random Forest[18], Logistic Regression[19], and Decision trees[20] in the Weka toolbox[21].

Table 1. shows the initial parameters for training the regression models. Figure 4 shows the dependency of the features extracted to train the data. It shows the alert count for a week. Figure 4(a) depicts the averaged alert count w.r.t. features extracted such as type of the day (Rainy/Non-Rainy, Windy/Non-Windy). Figure 4(b) depicts the pattern of alert count during different parts of the day such as early morning, late morning. Figure 4(c) shows the variation of averaged alert count w.r.t. temperature.



(a) (b) (c)

Date	Hour	Day/Night	Working/Holiday	Rainy/NRainy	Windy/NWindy	Humidity	Temperature	Alerts
01-07-2019	05:00-08:00	Early Morning	Working	Nrainy	Nwindy	64	20	10
01-07-2019	08:00-12:00	Late Morning	Working	Nrainy	Nwindy	63	20.4	24
01-07-2019	12:00-16:00	Afternoon	Working	Nrainy	Nwindy	61	20.1	19
01-07-2019	16:00-20:00	Evening	Working	Nrainy	Nwindy	62	19.8	34
01-07-2019	20:00-05:00	Night	Working	Nrainy	Nwindy	63	20	22

(d)

Humidity	Temperat	Early Mori	Late Morn	Afternoor	Evening	Night	Working	Holiday	Rainy	Nrainy	Windy	Nwindy	Alerts
64	20	1	0	0	0	0	1	0	0	1	0	1	10
63	20.4	0	1	0	0	0	1	0	0	1	0	1	24
61	20.1	0	0	1	0	0	1	0	0	1	0	1	19
62	19.8	0	0	0	1	0	1	0	0	1	0	1	34

(e)

Figure 4. (a) Average number of alerts w.r.t extracted features, (b) Number of alerts during day for a week, (c) Variation of number of generated alerts w.r.t temperature, (d) Server data displaying the extracted features, and (e) Training features for regression models after one-hot encoding.

Table 1. Training parameters for regression models

Regression model	Batch size	Max iterations	No. of decimal places	Ridge	Maximum depth	Num folds	Min number	Random state
Decision tree	100	1000	10	1.E-8	-	-	-	--
Logistic regression	100	1000	-	-	6	3	50	-
Random forest	100	1000	-	-	6	5	100	60

4.2 Time-Series Analysis of Data

To model short-term predictions on an hourly basis, the independent aspect of the data is analyzed. The raw data is preprocessed and converted to time-series data (as discussed in section 3.1). The statistical information, extracted from the training dataset is used as features to train SVR[22]. The features extracted from the data are: (a) Mean, (b) Weighted moving average, (c) Variance, (d) Standard Deviation, (e) Entropy. Figure 5 shows the values of features extracted from the time-series data. The training is done using a two-stage fusion approach as illustrated in Fig. 6. The outputs from the first stage are fed as input to the prediction model in the second stage. The final output of the trained model is the predicted number of alerts.

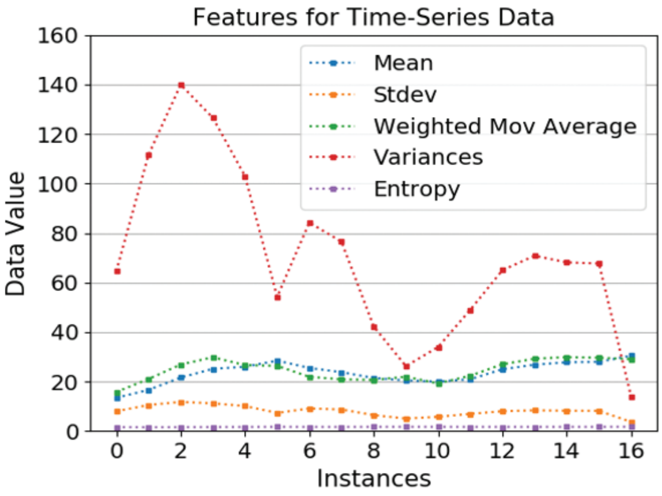


Figure 5. Extracted features for time-series analysis.

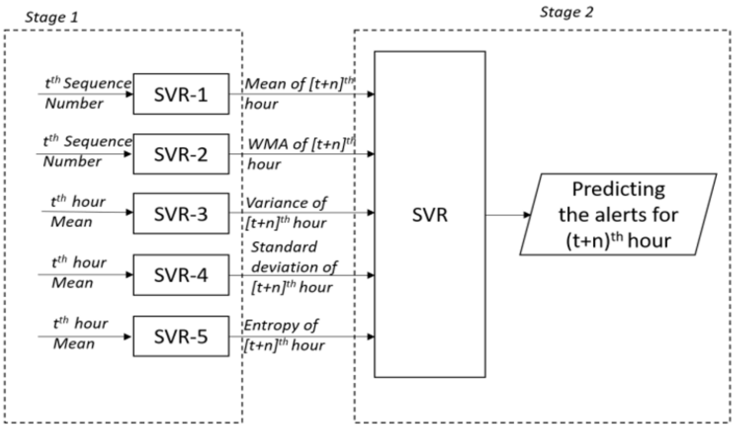


Figure 6. Two stage fusion approach for prediction of alerts n hours ahead of time.

It was observed that the input to SVRs is the data till tth hour while the output describes the alerts predicted at (t+n)th hour (where n=1,2,3,4…,24), the training data is prepared independently for every hour. As an instance, to predict the alerts of nth hour, the features from data with window size n are extracted with a sliding window of size 1. This way we are capturing the statistical information for every nth hour, which will then be used to predict the alerts for (t + n)th hour. So, we have trained 24 models for each hour of the day. To predict the number of alerts, the (t + n)th hour model is invoked and input is given as the sequence number of the next value in time-series data and the calculated mean for tth hour. The output will be the number of alerts in (t + n)th hour.

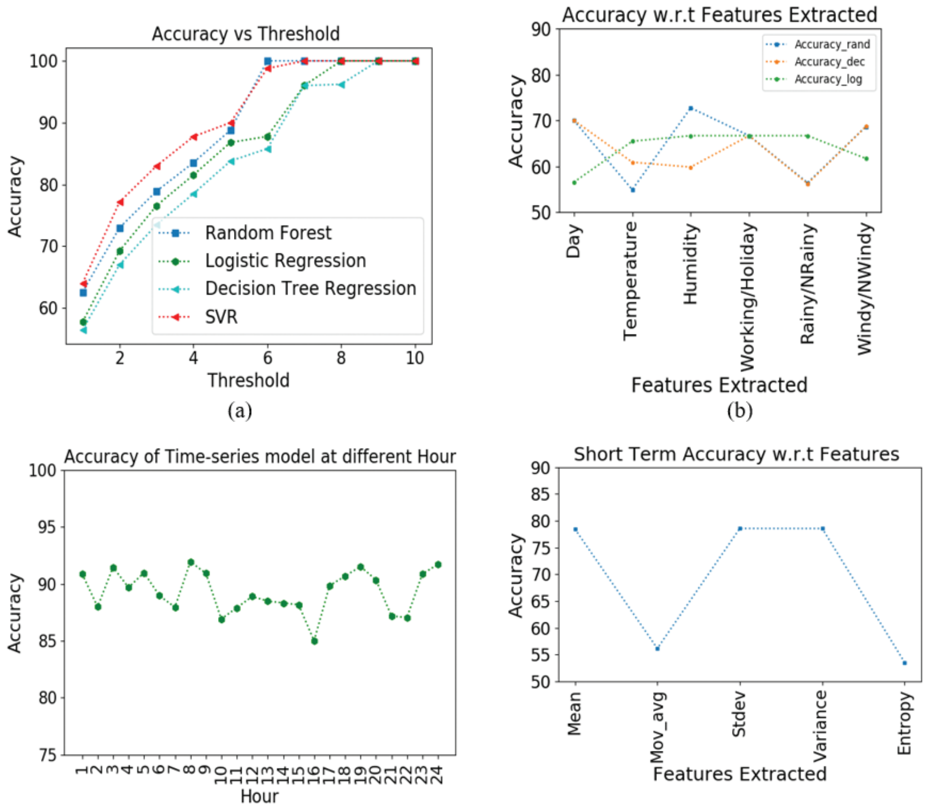
4.3 Detection of Anomalous Behavior

The models are tested on real-time data. The alerts are predicted automatically. Any deviation from the actual alert count and the predicted alert count results in generation of alerts for anomalous behaviour. A periodic thread runs these models to calculate the expected number of alerts in the next hour or the nth hour. As it is quite evident that the exact number of alerts does not match the real-time scenario. So, a threshold value of ‘5’ is defined, based on the results obtained. If the actual alerts do not lie in the range of expected threshold value, then alarms are generated depicting an anomalous behaviour.

5. RESULTS AND OBSERVATIONS

The proposed methodology was tested on real-time data. The feature-based model requires the historical data which was aggregated from July 2018 to December 2019. The models were trained on the feature set prepared by the historical data as well as scarped weather data. As, some of the features are binary like rainy or non-rainy day, weekend, or working day, one-hot encoding[23] is used to prepare the training dataset for those features. The training data consisted of 2740 data instances with 13 features after encoding (as shown in Fig. 4(e)) while the models were tested on data collected for a month consisting of 150 data instances. Figure 7(a) shows the accuracy w.r.t. the variable threshold value to determine the range in which the predicted value is considered to be accurate.

Figure 7(b) depicts the accuracy of regression models w.r.t individual features (long-term prediction) and Fig. 7(d) depicts the accuracy of SVR model w.r.t the individual features (short-term prediction). The accuracies are calculated considering the threshold values. It can be seen that when cumulative features are considered the accuracy of the models is increased (Fig. 7(a)(c)).



(a) (b) (c) (d)

Figure 7. (a) Long-term prediction accuracy w.r.t individual features (b) Accuracy w.r.t the threshold value (c) Short-term prediction accuracy w.r.t individual features (d) Cumulative accuracy of the proposed dual-stage model.

It can be seen that Random Forest has the highest accuracy of around 88% while logistic regression and decision trees have an approx. accuracy of 86 % and 83 %, respectively. The accuracies for all the models tend to be 100 % after a certain threshold value because of the wide window size to adjust the error in the models. Table 2 shows the performance parameters (root mean squared error (RMSE), mean square error (MSE), mean absolute error (MAE)) of the regression models at a threshold window of size 0.

Table 2. Performance parameters for regression models

Regression models	RMSE	MSE	MAE (degree)
Random Forest	4.3	20.39	3.74
Decision Tree	4.55	20.78	3.75
Logistic Regression	5.53	30.64	4.36

The time-series analysis is the continuous monitoring of the data. Hence, data of 90 days is used to predict the alerts in (t+n)th hour. 24 SVR models are prepared with variable window size of n= 1,2,3...24 for every hour. Radial based kernel (RBF) is used as the kernel function for training SVR. The hyperparameters (c, epsilon, gamma) at which the performance parameters (RMSE, MSE, MAE, Best Scoring) calculated for trained model is given in Table 3. Figure 7(c) shows the accuracy results for time-series analysis. 24 models were designed and tested for each hour. The data for 10 days was tested and a threshold value of '5' is considered. It can be seen that accuracy for the models is averaged around 89 %.

6. CONCLUSION

We have investigated the alerts generated by the perimeter intrusion detection systems. The deployed environment is studied to find the dependency of the generated alerts. It is helpful in making the long-term predictions of the alerts which will help in differentiating the actual alerts from the false alert situation. Different regression techniques are used and it is found that *Random Forest* offers better accuracy results. Further, the independent aspect of data is exploited by using the time-series analysis. A dual-stage approach to predict the short-term alerts is proposed using the support vector regression (SVR), which provides the good accuracy results. The short-term prediction is helpful in visualizing an alert situation during a day. The system is able to detect the anomalous behavior of alerts and warns the operators for discrepancy in the alerts generated and expected. It is also helpful in recognizing the alert flood situations based on the dependent parameters.

In the future, the system can also incorporate the component of device diagnostics to provide an overall scenario of alerts. It will be helpful in catering to the difference in the alerts generated in the case of faulty sensors.

Table 3. Hyper-parameters and performance parameters for SVR

Hour (n)	RMSE	MSE	MAE (degree)	Best scoring (neg_mean_ squared_error)	Parameter for SVR model		
					C	Epsilon	Gamma
1	1.35	1.83	1.01	-4.604	10	0.1	3
2	2.54	6.47	2.42	-7.095	1	3	1
3	1.57	2.46	1.07	-5.967	1	0.5	1
4	3.30	10.91	2.94	-14.120	1	5	3
5	0.10	0.01	0.1	-2.148	10	0.1	1
6	0.45	0.21	0.45	-2.233	10	0.5	0.5
7	4.07	16.63	3.99	-14.935	1	5	3
8	0.10	0.01	0.1	-2.671	10	0.1	1
9	0.47	0.22	0.47	-3.771	10	0.5	1
10	3.04	9.28	2.64	-15.317	1	5	5
11	0.01	0.0001	0.01	-3.987	10	0.01	3
12	1.35	1.83	1.01	-4.604	10	0.1	3
13	0.1	0.01	0.1	-5.606	10	0.1	3
14	0.09	0.1	0.009	-6.125	10	0.1	5
15	0.10	0.1	0.01	-6.478	10	0.1	5
16	3.29	10.85	2.81	-14.699	10	5	3
17	0.98	0.96	0.74	-7.545	10	0.5	5
18	1.31	1.73	0.58	-7.806	10	0.1	5
19	0.53	0.28	0.53	-9.142	10	0.5	5
20	3.14	9.91	2.96	-13.428	1	0.5	5
21	4.03	16.24	3.95	-13.566	1	0.5	5
22	4.34	18.91	4.21	-14.238	1	1	5
23	3.67	13.51	3.59	-16.463	10	5	5
24	3.65	13.32	3.5	-16.215	1	5	5

REFERENCES

1. Artificial intelligence is going to supercharge surveillance - The Verge. https://www.theverge.com/2018/1/23/16907238/artificial-intelligence-surveillance-cameras-security. [Accessed on: 18-Jun-2020].
2. Perimeter security uses advanced sensors to provide fail-safe intrusion detection. https://www.militaryaerospace.com/rf-analog/article/16718210/perimeter-security-uses-advanced-sensors-to-provide-failsafe-intrusion-detection. [Accessed on: 18-Jun-2020].
3. Chatfield, C. Time-series forecasting. CRC press, 2001.
4. Trainor, C.; Subramanian, G. & Vavrasek, D. Prediction of false alarms in sensor-based security systems. US patent 10359771. 23 July 2019.
5. Kapuschat, J.; Townley, T.; Joseph, M. & Yu, N. Predicting service for intrusion and alarm systems based on signal activity patterns. US patent 10380521. 13 Aug 2019.
6. Trundle & S.S. Alarm probability. US patent 9013294. 21 April 2015.
7. Adonailo, R.S.; Li, T.T. & Zakrewski, D.S. False alarm reduction in security systems using weather sensor and control panel logic. US patent 7218217. 15 May 2007.
8. Lahmiri, S. Minute-ahead stock price forecasting based on singular spectrum analysis and support vector regression. *Appl. Math. Comp.*, 2018, **320**, 444–451. doi: 10.1016/j.amc.2017.09.049
9. Hong, W.C. Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. *Neurocomputing*, 2011, **74**(12-13), 2096-2107. doi: 10.1016/j.neucom.2010.12.032
10. Hong, W.C. Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting. *Neural Comp. Applications*, 2012, **21**(3), 583-593. doi: 10.1007/s00521-010-0456-7
11. Hong, W.C.; Dong, Y.; Zheng, F. & Wei, S.Y. Hybrid evolutionary algorithms in a SVR traffic flow forecasting model. *Appl. Math. Comp.*, 2011, **217**(15), 6733-6747. doi: 10.1016/j.amc.2011.01.073
12. Fan, G.F.; Qing, S.; Wang, H.; Hong, W.C. & Li, H.J. Support vector regression model based on empirical mode decomposition and auto regression for electric load forecasting. *Energies*, 2013, **6**(4), 1887-1901. doi: 10.3390/en6041887
13. Geng, J.; Huang, M.L.; Li, M.W. & Hong, W.C. Hybridization of seasonal chaotic cloud simulated annealing algorithm in a SVR-based load forecasting model. *Neurocomputing*, 2015, **151**, 1362-1373. doi: 10.1016/j.neucom.2014.10.055
14. Hong, W.C.; Dong, Y.; Lai, C.Y.; Chen, L.Y. & Wei, S.Y. SVR with hybrid chaotic immune algorithm for seasonal load demand forecasting. *Energies*, 2011, **4**(6), 960-977. doi: 10.3390/en4060960
15. Xiang, Y.; Gou, L.; He, L.; Xia, S. & Wang, W. A SVR–ANN combined model based on ensemble EMD for rainfall prediction. *Applied Soft Computing*, 2018, **73,**

874-883.
doi: 10.1016/j.asoc.2018.09.018

16. Yu, P.S.; Yang, T.C.; Chen, S.Y.; Kuo, C.M. & Tseng, H.W. Comparison of random forests and support vector machine for real-time radar-derived rainfall forecasting. *J. Hydrology*, 2017, **552**, 92-104.
doi: 10.1016/j.jhydrol.2017.06.020
17. World Temperatures — Weather around the world. https://www.timeanddate.com/weather/. [Accessed on: 18-Jun-2020].
18. Liaw, A. & Wiener, M. Classification and regression by random Forest. *R news*, 2002, **2**(3), 18-22.
19. Menard, S. Applied logistic regression analysis, Sage, 2002.
20. Quinlan, J.R. Induction of decision trees. *Machine Learning*, 1986, **1**(1), 81-106.
doi: 10.1023/A:1022643204877
21. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P. & Witten, I.H. The WEKA data mining software: an update. *ACM SIGKDD Explor. Newsl.*, 2009, **11**(1), 10-18.
doi: 10.1145/1656274.1656278
22. Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Statistics Computing*, 2004, **14**(3), 199-222.
doi: 10.1023/B:STCO.0000035301.49549.88
23. Using categorical data with one hot encoding | Kaggle. https://www.kaggle.com/dansbecker/using-categorical-data-with-one-hot-encoding.[Accessed on: 18-Jun-2020].

CONTRIBUTORS

Ms Rizul Aggarwal received her BTech (Computer Science) from DAV Institute of Engg. and Technology, in 2015 and MTech (Computer Science) from PEC University of Technology, Chandigarh, in 2017. She is currently working as Scientist at Bharat Electronics Ltd., Ghaziabad, India.
In the current study, she was involved in development, testing, evaluation and validation of results and paper writing.

Ms Anjali Goswami received her BTech (Computer Science) from BTKIT, Dwarahat, in 2014 and MTech (Computer Science) from GBPANT, Pauri, in 2017.Currently working as a Scientist at Bharat Electronics Ltd., Ghaziabad, India.
In the current study, she was involved in the development, testing, evaluation and validation of results and paper writing.

Mr Jitender Kumar received his BTech (Computer Science) from College of Technology, Pantnagar, in 2003. Currently working as a Senior Scientist in Central Research Lab, Bharat Electronics Ltd., Ghaziabad, India.
In the current study, he has extended supervision to the main author for analysis, the outline of research work, and final scrutiny of the paper.

Mr G.A. Chullai received his BTech (Electronics & Communication) from KITS, Nagpur University, in 1999, and MTech (Fiber Optics) from IIT Kharagpur, in 2001. Currently working as a Senior Scientist in Central Research Lab, Bharat Electronics Ltd., Ghaziabad, India.
In the current study, he guided the entire study, interpretation of results, testing data and paper review.